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

**Peraton**

**www.peraton.com**

**Final Report**

**Date:**

12/03/2021

**Team Members:**

Rolando Gonzalez

Elshaday Alemayehu

Yusuf Siddiqui

Arial Carvalho

Eduardo Heredia

# Executive Summary

This document details the agreement between the sponsor and the development team. The purpose of this document is to provide a high-level overview of the development process needs and features. It focuses on the capabilities needed by the sponsor and why those needs exist. This document outlines the project plan, scope of the project, and deliverables. This document is subject to change as needed by the sponsor.

The project plan of the document presents a series of scheduled activities and important details for the development of the project. Some details include the project management plan, requirements specification, architecture design, and the testing plan. The tracking of this information is crucial for the successful deliverable of the project ensuring all three main pillars are met: budget, schedule, and scope.

The goal of this project is to create a web application designed to automate the process of analyzing contractual documents through the use of machine learning models and natural language processing techniques. These techniques are crucial in identifying key parts of the document that may indicate a positive or negative underlying sentiment. This helps to identify whether a contract contains polarizing words that are favorable or unfavorable to either party in a contractual agreement.

Table of Contents

[Executive Summary 2](#_Toc89416654)

[List of Figures 5](#_Toc89416655)

[List of Tables 6](#_Toc89416656)

[1. Introduction 7](#_Toc89416657)

[1.1 Purpose and Scope 7](#_Toc89416658)

[1.2 Product Overview 7](#_Toc89416659)

[1.3 Structure of the Document 7](#_Toc89416660)

[1.4 Terms, Acronyms, and Abbreviations 8](#_Toc89416661)

[2. Project Management Plan 9](#_Toc89416662)

[2.1 Project Organization 9](#_Toc89416663)

[2.2 Lifecycle Model Used 10](#_Toc89416664)

[2.3 Risk Analysis 11](#_Toc89416665)

[2.4 Hardware and Software Resource Requirements 13](#_Toc89416666)

[2.5 Deliverables and Schedule 14](#_Toc89416667)

[2.6 Monitoring, Reporting, and Controlling Mechanisms 15](#_Toc89416668)

[2.7 Professional Standards 16](#_Toc89416669)

[2.8 Evidence All the Artifacts Have Been Placed Under Configuration Management 17](#_Toc89416670)

[2.9 Impact of the Project on Individuals and Organizations 18](#_Toc89416671)

[3. Requirement Specifications 19](#_Toc89416672)

[3.1 Stakeholders for the System 19](#_Toc89416673)

[3.2 Use Case Model 22](#_Toc89416674)

[3.2.1 Graphic Use Case Model 22](#_Toc89416675)

[3.2.2 Textual Description for Each Use Case 23](#_Toc89416676)

[3.3 Rationale for Your Use Case Model 26](#_Toc89416677)

[3.4 Non-functional requirements 26](#_Toc89416678)

[4. Architecture 28](#_Toc89416679)

[4.1 Architectural Style(s) Used 28](#_Toc89416680)

[4.2 Architectural Model 29](#_Toc89416681)

[4.3 Technology, Software, and Hardware Used 30](#_Toc89416682)

[4.4 Rationale for Your Architectural Style and Model 31](#_Toc89416683)

[5. Design 32](#_Toc89416684)

[5.1 GUI (Graphical User Interface) Design 32](#_Toc89416685)

[5.2 Static Model – Class Diagrams 35](#_Toc89416686)

[5.3 Dynamic Model – Sequence Diagrams 36](#_Toc89416687)

[5.4 Rationale for Your Detailed Design Model 37](#_Toc89416688)

[5.5 Traceability From Requirements to Detailed Design Model 38](#_Toc89416689)

[6. Test Plan 41](#_Toc89416690)

[6.1 Requirements/Specifications-based System Level Test Cases 41](#_Toc89416691)

[6.2 Traceability of Test Cases to Use Cases 47](#_Toc89416692)

[6.3 Techniques Used for Test Generation 48](#_Toc89416693)

[6.4 Assessment of the Goodness of Your Test Suite 49](#_Toc89416694)

[Acknowledgment 52](#_Toc89416695)

[References 53](#_Toc89416696)

# List of Figures

Figure 1: GitHub Configuration Management 17

Figure 2: Stakeholder Diagram 21

Figure 3: System Use Case 22

Figure 4: Architectural Model 29

Figure 5: Login Screen 32

Figure 6: Normal User Main Page 32

Figure 7: Normal User Document Results 33

Figure 8: Administrator Main Page 33

Figure 9: Administrator Document Results 34

Figure 10: Static Model - Class Diagram 35

Figure 11: Account Sequence Diagram 36

Figure 12: Upload Sequence Diagram 36

Figure 13: Test 1 - File Upload Test 49

Figure 14: Test 1 - Result 49

Figure 15: Test 2 - Verify Valid Upload 49

Figure 16: Test 2 - Result 50

Figure 17: Test 3 - Positive polarity test 50

Figure 18: Test 3 - Result 50

Figure 19: Test 4 - Non-positive polarity test 51

Figure 20: Test 4 - Result 51

Figure 21: Test 5 - Negative polarity test 51

Figure 22: Test 5 - Result 51

# List of Tables

Table 1: Acronyms/Abbreviations 8

Table 2: Project Organization 9

Table 3: Issues and Risks 11

Table 4: Deliverables and Schedule 14

Table 5: Volere Stakeholder Analysis 19

Table 6: Requirements Traceability 38

Table 7: Login-1 41

Table 8: Login-2 42

Table 9: Upload-1 43

Table 10: Upload-2 43

Table 11: Contract Favorability Skew-1 44

Table 12: Contract Favorability Skew-2 45

Table 13: View ML Model performance-1 45

Table 14: View ML Model performance-2 46

Table 15: Traceability of Test Cases to Use Cases 47

# 1. Introduction

## 1.1 Purpose and Scope

The purpose of this project is to design a machine learning model with the capabilities to improve the decision making process for business analysts and project managers when analyzing contracts by identifying key areas that may have bias towards one entity. The scope of the project is a program with capability of determining the quality and maturity of documents based on predetermined hypotheses. This product is designed for analyzing contractual documents between entities such as SOW (statement of work) documents.

The scope is to ensure the project pertains to the requirements and objectives of the work required to complete the project successfully. The scope also states the methodology we are applying to manage the scope of the project, determining the roles and responsibilities for the project, and how we are going to validate and control the scope of the project. The project sponsor is responsible for defining the scope of the project, managing all the modifications related to scope, and approving the deliverables for scope changes.

## 1.2 Product Overview

The product to be developed is a web application with capability of determining the quality and maturity of documents based on predetermined hypotheses. The program shall read and process documents using NLP technologies, and accomplish its goal by using ML/AI models. The underlying motivation for this product is to improve decision-making process as determining the level of risk a contract may impose.

This product is designed to automate the task of analyzing contractual agreements between two entities and determine the favorable and unfavorable aspects within the document so that a business analyst and project manager can speed up the process of discovering whether additional negotiations need to be discussed.

## 1.3 Structure of the Document

The structure of this document follows the main steps of the development process starting from creating a project management plan to ensure the project scope is identified and the needs of the sponsor are communicated to the development team. The next part covers requirements elicitation to ensure the project pertains to the requirements and objectives of the work required to complete the project successfully. The next part covers the architectural 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. The design section covers the prototypes, the look and feel, and the models. Finally, the testing section will highlight the types of test for quality assurance of the product.

## 1.4 Terms, Acronyms, and Abbreviations

Table : Acronyms/Abbreviations

|  |  |
| --- | --- |
| **Acronyms/Abbreviations** | **Meaning** |
| ML | Machine Learning |
| AI | Artificial Intelligence |
| NLP | Natural Language Processing |
| UX | User Experience |
| RAID | Risk, Assumptions, Issues, Dependency |
| SDLC | Software Development Life Cycle |
| WBS | Work Breakdown Structure |
| OSX | Macintosh Operating System X |
| UI | User Interface |
| DAO | Data Access Object |
| CRUD | Create, Read, Update, Delete |
| GUI | Graphical User Interface |
| SOW | Statement of Work |

# 2. Project Management Plan

## 2.1 Project Organization

Table : Project Organization

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Role** | **Responsibilities** |
| Rolando Gonzalez | Project Lead | 1. Be the representative and intermediary between the team and the sponsors 2. Ensure team is meeting weekly 3. Keep track of activities performed and deliverables 4. Communicate project stakeholders and ensure deliverables are meeting deadlines |
| Ariel Carvalho | ML/AI Developer | 1. Design ML models 2. Research and implement ML/AI algorithms 3. Select and test data set 4. Collect and verify ML model results are accurate |
| Elshaday Alemayehu | NLP Specialist | 1. Design NLP systems 2. Use effective text representation to transform natural language into useful features 3. Implement accurate algorithms and tools for NLP tasks 4. Test and validate NLP requirements |
| Yusuf Siddiqui | ML/AI Developer | 1. Design ML models 2. Research and implement ML/AI algorithms 3. Select and test data set 4. Collect and verify ML model results are accurate |
| Eduardo Heredia | Front-end/UX Developer | 1. Create prototypes and wireframes 2. Implement and test user-friendly design 3. Develop responsive user interfaces 4. Validate input and output from users |

## 2.2 Lifecycle Model Used

## 

Development Lifecycle Model

Our project would be using a Waterfall life cycle model that involves the following major steps:

* Requirement

The team will be able to analyze the requirements to fully understand the problem and document them properly.

* Design

In this phase, the team will take the requirement specification and the system design will be prepared. This system design helps us in specifying hardware and system requirements.

* Implementation

In this phase, we will begin technical implementation based on the system design.

* Testing

All the units developed in the implementation step will be integrated into a system after testing each unit.

* Deployment

Once the functional and non-functional testing is done,  the product will be deployed in the Peraton production environment.

Machine Learning Lifecycle Model

Our project would be using Machine learning life cycle that involves the following major steps:

* Gathering Data
  + In this first step of the machine learning life cycle, the team will identify different data sources that can be collected from different sources such as files, documentation, and company contracts. By integrating the data obtained from different sources, the team would be able to get a coherent set of data.
* Data Preparation
  + After getting datasets, we put our data in the right place and prepare it to use in our machine learning training. This can be done using data exploration to understand the format and characteristics of the data for a better understanding of the data that can lead to an effective outcome.
* Data Processing
  + In this ML lifecycle step, our team will be cleaning and converting raw data into a usable format using natural language processing techniques.
* Analyze Data
  + The team will build machine learning models to analyze the data using different machine learning techniques and evaluate the model.
* Train and Test the Model
  + The team will use datasets to train the model using different machine learning algorithms approaches. This step will help us to understand the various patterns and features within our data. After training, we will determine the percentage accuracy of the model by testing against the trained data.
* Deployment
  + Finally, we will deploy the model in the real-world system.

## 2.3 Risk Analysis

Table : Issues and Risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| High Risk |  |  |  |  |  |
| Med Risk | **Project - Issues and Risks** | | | | |
| Low Risk | **Item #** | **Description** | **Resolution Plan / Mitigation Plan** | **Contingency Plan** | **Risk Owner** |
|  | R-1 | Unplanned work that must be accommodated | Attend project scheduling meetings.  Document all assumptions made in planning and communicate to the project manager before project kick off. | Escalate to the team leader with plan of scheduling, including impact on time, cost, and quality | Team Leader |
|  | R-2 | Project may experience delays due to code development components being built from scratch. This may consume time and effort, compared to reusing codes that already exist. | appoint more resources to the developing team.  Code reuse, developers are able to use stable and trusted code libraries, best for building blocks  Open Source | using Open Source for building block in implementation of code | Developers |
|  | R-3 | Project may experience delay due to lack of expertise in using test frameworks. This might be a challenge for software testers to complete their task in time. | Hand the task to team members with more testing skills to learn more about the required test framework to help with the task | Give team members open source learning resources to develop such skill in short time | Developers |
|  | R-4 | Project may experience delays due to incompatibility issues that might arise on production environments which the development teams haven’t encountered on non-production environments. | Involve experienced devOps engineers to help prepare for these kinds of problems  Replicate the issues on non-production environments to help mitigate the issues | Communicate with production environment support team for possible solution to the incompatibility problems | Developers |
|  | R-5 | Project schedule is not clearly defined or understood | Share the schedule and go through upcoming tasks at  each weekly project progress meeting. | Revisit the schedule with team leader and relaunch the new project schedule | All Team members |
|  | R-6 | Lack of communication, causing lack of clarity and confusion. | Write a communication plan that includes: the frequency, goal, and audience of each communication.  Identify stakeholders early and make sure they are considered in the communication plan. | Correct misunderstandings immediately.  Clarify areas that are not clear swiftly using assistance from Project Sponsor if needed. | Team lead |
|  | R-7 | Project might be delayed due to late arrival of necessary documentation from stakeholders | Set up a meeting with stakeholders to discuss details about necessary resources and documentation | Start on other tasks that are not dependent on the specific documentations | Team lead |
|  | R-8 | Project design and deliverable definition is incomplete | Define the scope in detail via meetings and include inputs from subject matter expertise | Document assumptions made and associated risks | Project Sponsor |

## 2.4 Hardware and Software Resource Requirements

Hardware:

* Windows laptop environment with Wi-Fi connectivity

Software:

* Web browser (Chrome or Microsoft Edge preferred)
* Python (and the following libraries)
  + NLTK (Natural Language Toolkit)
  + TextBlob
  + Python-docx
  + PDFminer
  + Scikit-learn
* Jupyter Notebook
* Anaconda
* Flask and Django frameworks [6]
* MongoDB and PyMongo for database integration

New software/hardware each team member learned during the project:

* Yusuf Siddiqui
  + Learned about text extraction for pdf and Microsoft word files with (PDFminer) and (Python-docx)
  + Learned how to score sentiment with existing libraries (TextBlob)
  + Learned integration of backend and frontend using (Django)
* Rolando Gonzalez
  + Learned about Django: first time implementing this framework for our web application
  + Learned about Python: how to develop the frontend using Python
  + Learned about Jupyter Notebook: how ML and NLP is used to extract information from documents via Jupyter Notebook
* Elshaday Alemayehu
  + Learned about TextBlob: how to classify using pre-trained data on TextBlob
  + Learned about NLTK: how to find if a text is negative or positive and process text using NLTK
  + Learned about Pytest: how to write test cases using Python
* Eduardo Heredia
  + Learned about Django: creating the frontend of the web application using the Django framework
  + Learned about Jupyter Notebook: web-based interactive development environment
  + Learned about PyMongo: integration of MongoDB using PyMongo
* Ariel Carvalho
  + Learned about PyMongo: native driver to connect and manage records in MongoDB using Python
  + Learned about Flask: light-weight Python web framework to handle http requests between the back and frontend
  + Learned about Postman: platform that simulates http requests, simplifying and reducing duration of API development

## 2.5 Deliverables and Schedule

Table : Deliverables and Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Activity** | **Dependency** | **Completion Date** | **Team Members** |
| A01 | Project Management Plan | N/A | 09/10/2021 | All |
| A02 | Requirements Gathering | A01 | 09/24/2021 | All |
| A03 | Architecture Documentation | A02 | 10/08/2021 | All |
| A04 | Design Documentation | A03 | 10/22/2021 | All |
| A05 | Data Gathering | A02 | 10/29/2021 | Rolando Gonzalez |
| A06 | Developing Algorithms for Text Extraction | A05 | 11/08/2021 | Elshaday Alemayehu,  Yusuf Siddiqui |
| A07 | Data Processing(clean/prepare data) with NLP | A06 | 11/10/2021 | Elshaday Alemayehu, Yusuf Siddiqui |
| A08 | Classification | A07 | 11/12/2021 | Yusuf Siddiqui, Elshaday Alemayehu |
| A09 | Training ML Model | A08 | 11/12/2021 | Yusuf Siddiqui |
| A10 | Testing Plan | A09 | 11/12/2021 | All |
| A11 | Evaluating ML Model | A10 | 11/22/2021 | Yusuf Siddiqui  Eduardo Heredia |
| A12 | Integration of backend with frontend | A11 | 12/01/2021 | Yusuf Siddiqui, Eduard Heredia |
| A13 | Evaluating test cases | A12 | 12/02/2021 | Elshaday Alemayehu |
| A14 | Final Report | A13 | 12/03/2021 | All |

The project followed the above schedule to fit both of the chosen lifecycle models used in the project. This includes the deliverables required and the high-level milestones needed to successfully complete the project.

## 2.6 Monitoring, Reporting, and Controlling Mechanisms

* To set up our project monitoring and control process, we first are going to establish the project baselines such as scope, schedule, and budget. Using this information team members shall create a work breakdown structure (WBS) to break down the project into small units of work and sub-tasks. WBS will help us detect issues at an early stage which will be important to keep the project under control.
* The team will be using a requirement traceability matrix to ensure all requirements are defined, accepted requirements are broken down into development tasks and changes at any time during the development lifecycle are traced.
* The team will be using RAID to identify project risk, assumptions, issues, and dependencies. RAID will help us proactively manage the project.
* To monitor and control the project, we will be using project schedules to help us assist with project tracking, reporting progress, identifying task relationships, monitoring progress, and identifying issues early.
* Governance Management Plan

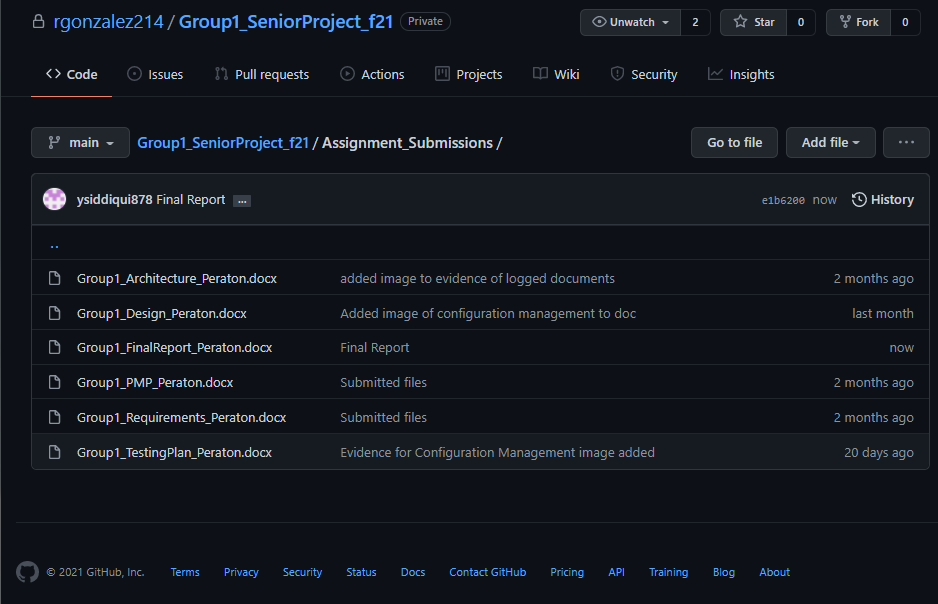
## 2.7 Professional Standards

* Each team member will accept full responsibility for their own work.
* Each team member will place great effort on the product and their related work to meet high quality.
* Each team member will support other team members.
* Each team member shall use their professional judgment to make informed decisions on their work.
* Each team member shall produce their own original work.
* The team shall display the best interest to their employer (Peraton).
* Team members missing meetings, deliverables, etc., will need to provide valid reasons to be excused.
  + A valid reason as an example:
    - Illness, family member’s death, travels, academic reasons, etc.
  + An invalid reason as an example:
    - “I forgot,” “device is drained,” “I am busy with something else,” etc.
* Team members who show misbehavior will be evaluated by the team and instructor. (This refers to appendix A)
  + 1st occurrence, the situation will be evaluated and the team shall resolve the underlying problem.
  + 2nd occurrence, in addition to the before, the instructor will be notified about the problem, and the team will meet with the instructor to resolve the problem.
  + 3rd occurrence, the professor will be notified, and the same conduct will occur. In addition, the team will be allowed to evaluate and decide whether the team member should be removed.
  + Each occurrence will be documented.
  + Examples of misbehavior may include but not limited to:
    - Missing deliverables, missing team meetings, poor quality of work, disrespectful attitude, harassment, unprepared for meetings, neglectful acts, etc.

All team members are subject to the above standards to ensure the project progresses smoothly and failure to comply with these rules will result in documentation of the incident and notifying the instructor if appropriate.

## 2.8 Evidence All the Artifacts Have Been Placed Under Configuration Management

Figure : GitHub Configuration Management



## 2.9 Impact of the Project on Individuals and Organizations

This project can help on an individual level by allowing for business analysts and project managers allowing for easier identification of bias towards an entity in a contract and will help speed up negotiations between clients and project managers.

The web application developed for Peraton will be a great supportive tool for the decision making process. Because of the functionalities it provides for analyzing risks imposed by contractual documents, decision makers are able to evaluate documents in a more effective way. Ultimately, the web application will have a financial and time saving impact as it can help the company to save money and time management as it can reduce the duration of accepting and rejecting a contract.

# 3. Requirement Specifications

## 3.1 Stakeholders for the System

Table : Volere Stakeholder Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Volere Stakeholder Analysis | | | | |
| Stakeholder Class | Stakeholder Role | Stakeholder Name | Stakeholder Rationale | Necessary Involvement |
| Core Team Member | System Architect | Yusuf Siddiqui | Designing ML model, classifier, and integration with frontend | Daily |
| Core Team Member | Project Manager | Rolando Gonzalez | Be the representative and intermediary between the team and the sponsors | Daily |
| Core Team Member | Testing Engineer | Elshaday Alemayehu | Designing ML model and performing testing | Daily |
| Core Team Member | Integration and QA Engineer | Eduardo Heredia | Integration, validation, and quality assurance | Daily |
| Core Team Member | Database Administrator | Ariel Carvalho | Designing database schema of system | Daily |
| Client | Director | Dave Gibson | Communicating requirements of system and setting scope of project | Weekly |

**Stakeholders Levels**

Level One: Stakeholders closely involved with the creation and development of the project

* Project Manager
* Testing Engineer
* System Architect
* Integration Engineer
* QA Engineer
* DB Administrator

Level Two: Stakeholders who provide leadership management and support

* Peraton Senior Managers
* Peraton IT support
* Peraton Director

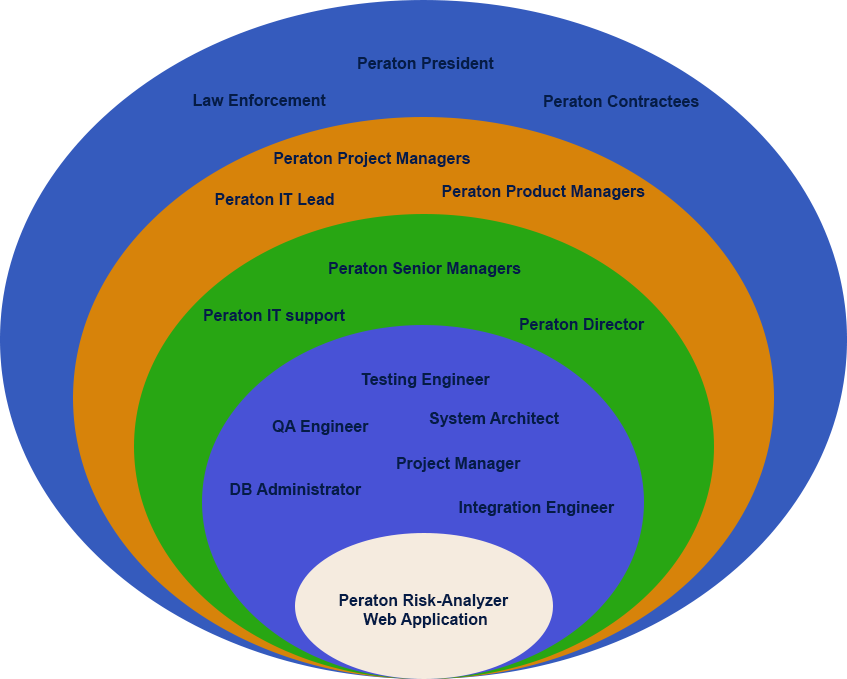
Level Three: Peraton employees who interact closely with the solution

* Peraton Project Managers
* Peraton IT Lead
* Peraton Product Managers

Level Four: External Stakeholders

* Peraton President
* Law Enforcement
* Peraton Contractees

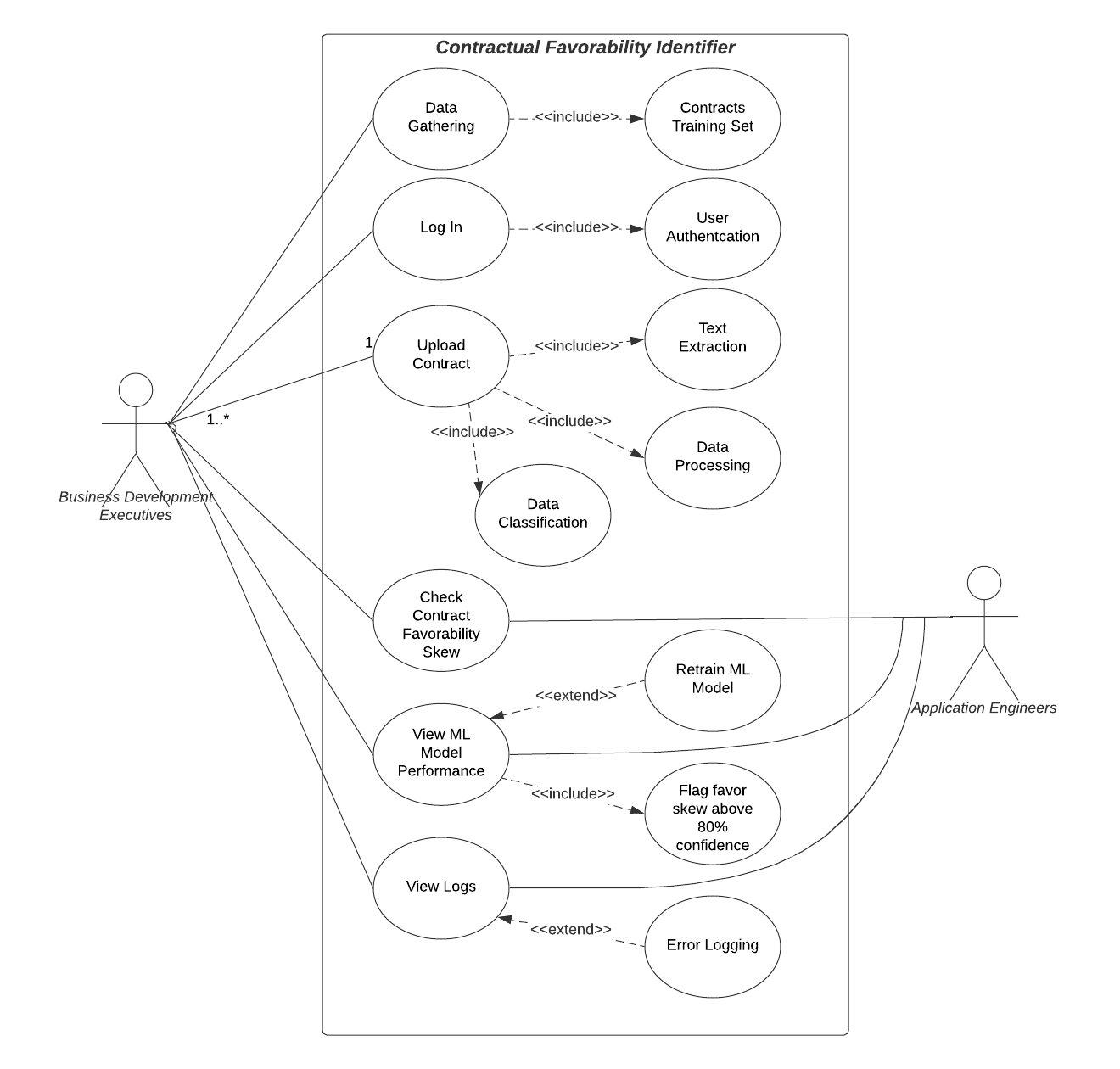
Figure : Stakeholder Diagram



## 3.2 Use Case Model

### 3.2.1 Graphic Use Case Model

Figure : System Use Case



### 3.2.2 Textual Description for Each Use Case

R01: **Account Creation**

Participating Actors: Business Development Executive

Entry condition: User has/knows an email address, password, and name

Flow of events:

1. User clicks on create account.
2. System displays create an account page.
3. User enters their account information.
4. System validates user information.
5. System persistently stores user data to the database.
6. System displays a message that an account has been created.

Exit conditions: User creates an account to access the system

Exceptions:

1. User clicks on create account.
2. System displays sign up page.
3. Users enter their information.
4. System identifies entered data that is already stored in the database.
5. System displays a message that an account with the data provided is in use.

Special Requirements: Once all required information is entered by the user, the system shall validate and displays the result within 5 seconds.

R02: **Log In**

Participating actors: Business Development Executive

Entry condition: User has an existing account created within the system.

Flow of events:

1. User clicks on log in
2. System displays login page
3. User enters their login information
4. System validates user information
5. System displays main menu from the dashboard

Exit conditions: User is granted access to the system main menu.

Exceptions:

1. User clicks on log in
2. System displays login page
3. User enters their information
4. System identifies entered data is not stored in database or incorrect
5. System displays a message that user has entered an incorrect login credential

Special Requirements: Once all required information is entered by the user, the system shall validate and displays the result within 5 seconds

E01: **User sets system to their preferred language**

Participating Actors: Business Development Executive

Entry condition: User has an option to choose from multiple languages

The flow of events:

1. User logs in to account
2. System asks the preferred desired language
3. System asks for document input
4. User enters information
5. System displays result in user preferred language

Exit conditions: System sets the chosen language as default for the user

Exceptions:

1. User logs in to their account
2. The system sets the default language to the user chosen language

Special Requirements: After the user chooses the language, the results should be converted to the preferred language within 5 seconds.

E02: **Upload Contract** - User shall be able to upload desired contract in pdf file format

Participating Actors: Business Development Executive

Entry condition: Users will be able to input the contract in the pdf file format

The flow of events:

1. User logs in to account
2. System asks to input desired document
3. The user enters pdf format documents
4. System checks if the document is in the correct format
5. The system processes the contract
6. System displays results

Exit conditions: The user enters a different document format

Exceptions:

1. User logs in to their account
2. System asks to input desired document
3. The user enters documents
4. System checks if the document is in the correct format
5. The system notifies the user to enter the correct format

Special Requirements: The system should be able to verify the document type in less than 3 seconds

E03: **Text Extraction** - The system must be able to extract text from user input documents

Participating Actors: Business Development Executive, System

Entry condition: In order to categorize the given contract into relevant dataset, the system will be able to complete text extraction from a given document

The flow of events:

1. Ask the user for input
2. The user enters pdf format documents
3. The system extracts text from the pdf file
4. The system cleans the documents
5. The system prepares text for a further use

Exit conditions: Text is extracted and organized properly

Exceptions: User aborts the process

Special Requirements: The system should be able to extract text from the document within minutes

E04: **Upload contract** - Data processing

Participating Actors:  Business Development Executive, Application Engineers

Entry condition: The system will be able to clean and convert raw data into a usable format

The flow of events:

1. System cleans text and split it into word
2. System processes the data
3. System categorizes the raw data into datasets
4. System stores the contract data into a dataset for further use

Exit Conditions: Raw data is converted to an applicable Machine Learning format

Exceptions: System not able to find keywords from the given document

Special Requirements: The system should be able to process the document within hours

Y01: **Check Contract Favorability Skew** – User shall be able to view flagged keywords and phrases

Participating Actors: Business Development Executive, Application Engineers

Entry condition: The ML model shall determine which keywords and phrases and provide insight on the favorability of flagged information

The flow of events:

1. User tunes acceptable parameters for ML model
2. User inputs document
3. System extracts text from the document
4. System processes the data
5. System classifies the data
6. System organizes data into charts and tables and displays information on the dashboard

Exit conditions: The user leaves parameters blank for ML model

Exceptions: User terminates the process early

Special Requirements: The system should be able to process the document within hours

Y02: **View ML model performance** – User shall be able to view overall performance of the ML model

Participating Actors: Business Development Executive, Application Engineers

Entry condition: The ML model shall provide information regarding words that did not meet the threshold parameters specified by the user.

The flow of events:

1. User enters bounds for confidence level and modifies settings
2. User inputs document
3. System provides information within specified bounds

Exit conditions: The user leaves parameters blank

Exceptions: User enters invalid bound range

Special Requirements: The system should be able to retrieve information within specified bounds

Y03: **View Logs** – System shall provide logs regarding information about the application such as successful logins, failed logins, errors, timestamps, warnings, and critical failures.

Participating Actors: Business Development Executive, Application Engineers

Entry condition: User is a valid elevated user within the system and has successfully logged in

The flow of events:

1. System prompts user for login information, title, employee id
2. User enters information correctly
3. User requests to view system logs
4. System provides the user with logs within an entered range (date, time, etc.)

Exit conditions: User does not have permission to access system logs

Special requirements: The system should be able to store and maintain logs for the user pertaining to the past 1 year.

## 3.3 Rationale for Your Use Case Model

The system should be able to parse and extract data to fit a ML model and should allow for retraining of data to be able to adapt to poor accuracy. The system shall also be able to provide detailed analysis of the requested information in the form of graphs and tables. The system shall be able to provide system logs to allow for monitoring and debugging.

## 3.4 Non-functional requirements

Security

* Passwords must be hashed and stored securely in the database
  + The application will not be available to all users, valid users must authenticate and verify identity with their title/position, employee id (if applicable), and password
* Sensitive information shall be encrypted for in-flight data
  + Information contained within the contracts that the program works with may be sensitive and will require some form of encryption when the application is run over a network
* System will automatically sign out within 15 minutes if no work is in progress and no response from the user is detected
  + Login session will be terminated to prevent sensitive data from being leaked when users are no longer using the application and no work is running
* Passwords must contain at least 8 characters including at least one special character, upper case letter, and number
  + Valid users will need to create a strong password as the program will deal with sensitive information
* Password delay after 10 consecutive failed login attempts
  + Too many failed login attempts will require the user to verify their identity or wait a certain amount of time to prevent brute-force attempts
* The user account database must use column-level transparent data encryption or better
  + A database that holds login information is sensitive and needs to be adequately encrypted in order to be properly protected
  + Require every column in the database a password for authorizing access to column data

Maintainability

* The system shall allow elevated users to provide a new training set of contractual documents
  + When there is high deviation from the expected outcome and the ML model’s accuracy is performing poorly, the model may need to be retrained with a new set of documents

Usability

* The system must work on OSX, Windows, and Unix-based systems
  + Even though development is done in a Windows environment, the dashboard should still be usable in other operating system environments
* The system may support multiple languages
  + A possible add-on would be to support contracts that may be in other languages
* The system must provide sorting functionality
  + Sorting should be provided for highest confidence levels, favorable/unfavorable keywords, etc.

Availability

* The system must retrieve contract documents one at a time
  + A user will be able to input one pdf document at a time instead of allowing users to input multiple files at a time.
  + Limit user input to only one at a time
* The system should not take longer than 3 seconds to log in the user
  + The system should validate user log-in by cross checking email and password and this should be complete in less than 3 seconds

The system must be able to process contract analysis within hours

# 4. Architecture

## 4.1 Architectural Style(s) 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

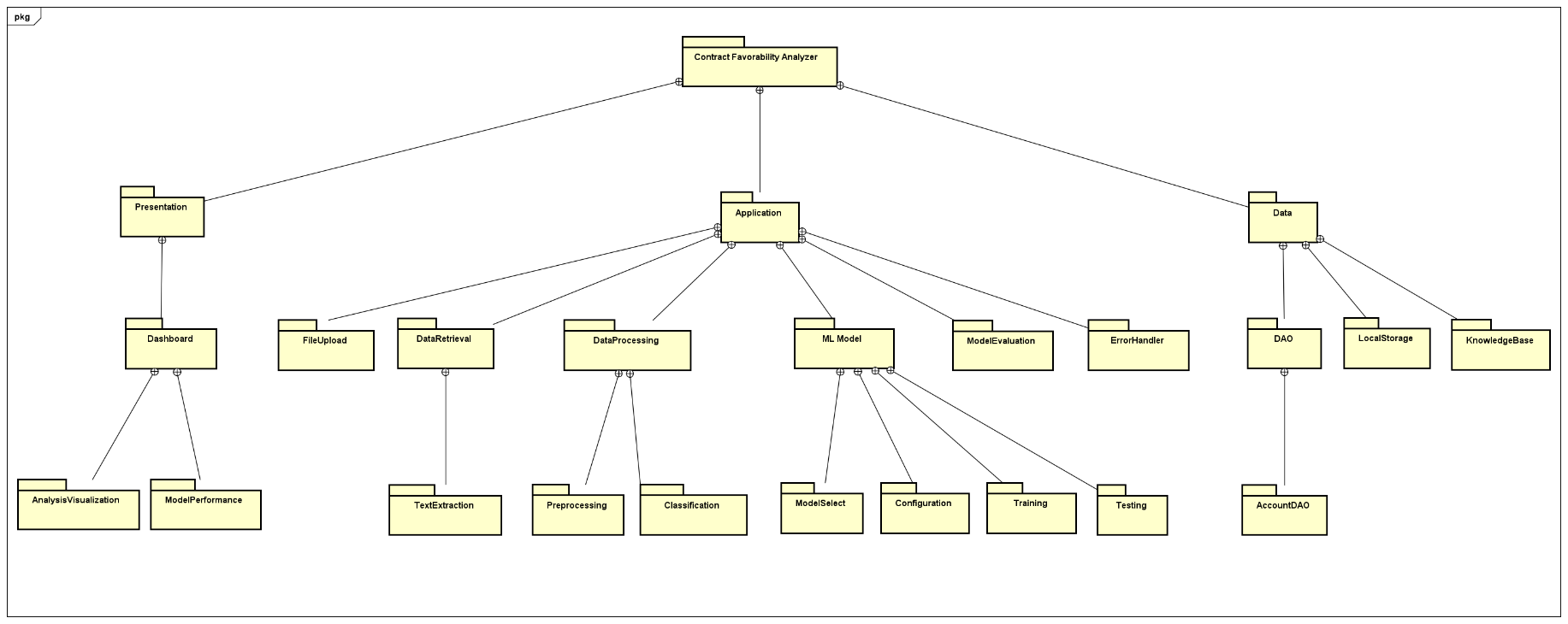
The DAO pattern is a structural pattern that isolates the operations (Create/Retrieve/Update/Delete) on the data from the business tier. For this system there will be user’s data to be stored on the database. The DAO class will handle any incoming/outcome to the database..

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

## 4.2 Architectural Model

Figure : Architectural Model



## 4.3 Technology, Software, and Hardware Used

**Text Extraction:**

* Python-docx: python library that will be used for extraction words in paragraphs from Microsoft Word documents [1].
* PDFminer: 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.
* TextBlob – Sentiment analyzer that uses a weighted average to score words in text and determine the polarity [5].

**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.

## 4.4 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.

# 5. Design

## 5.1 GUI (Graphical User Interface) Design

Figure : Login Screen

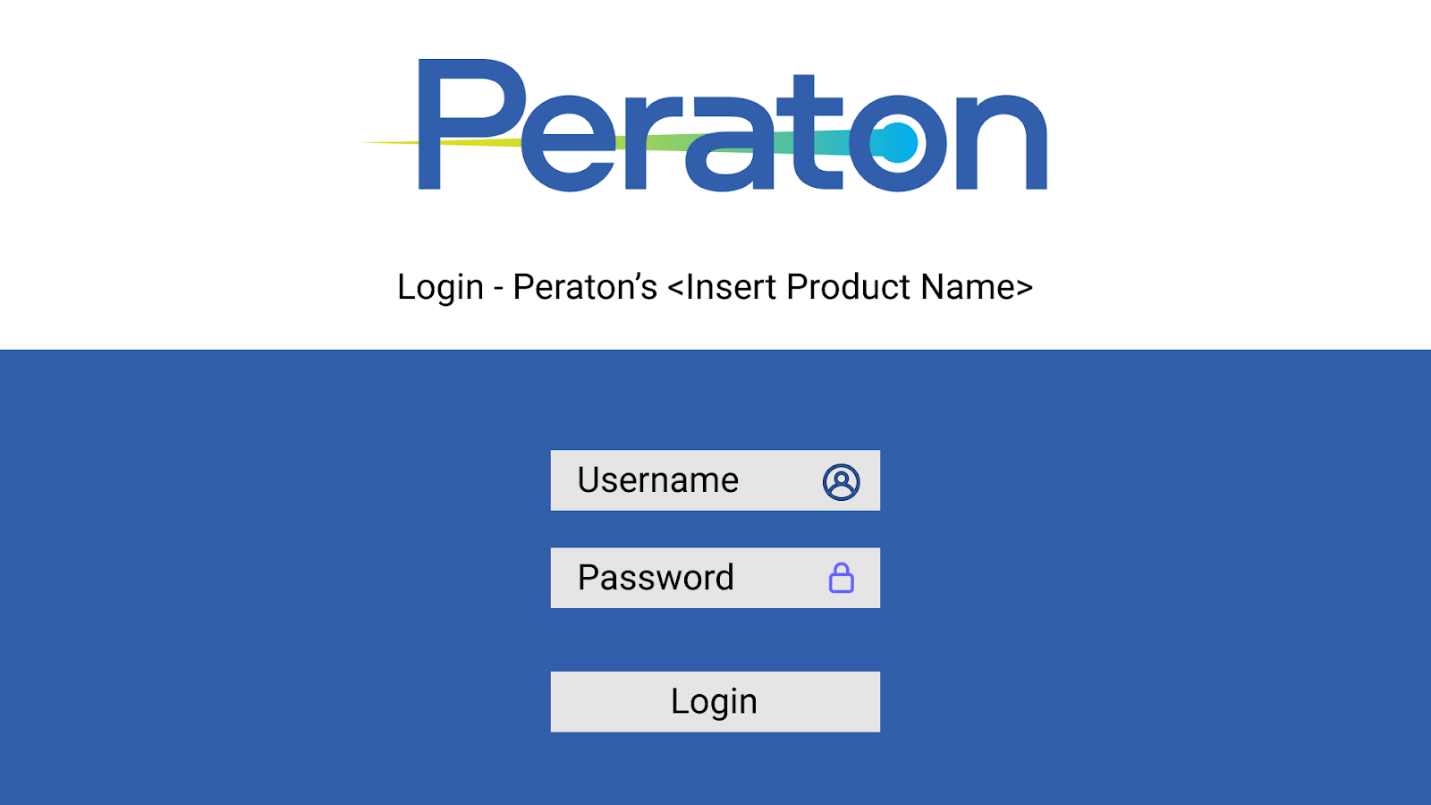


Figure : Normal User Main Page

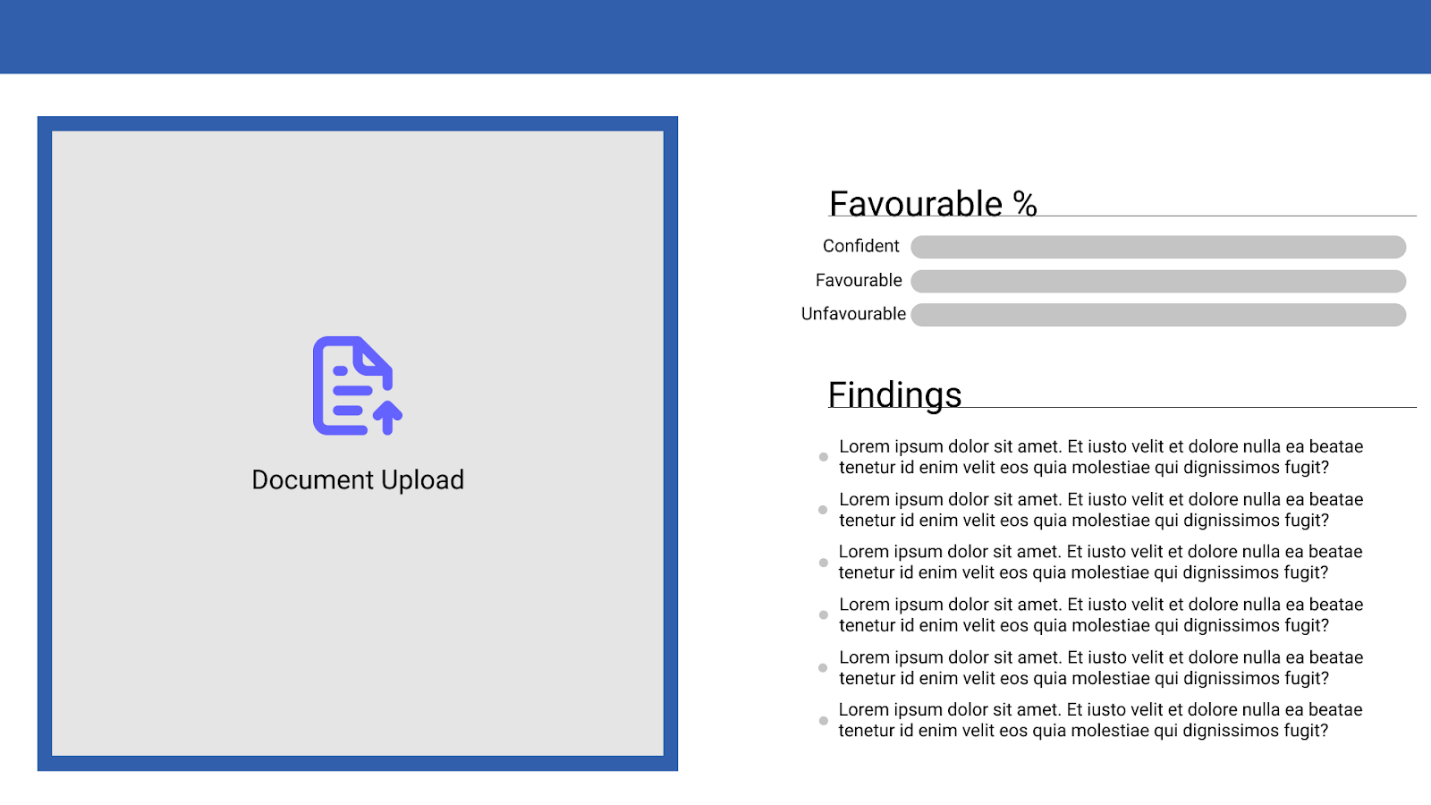


Figure : Normal User Document Results

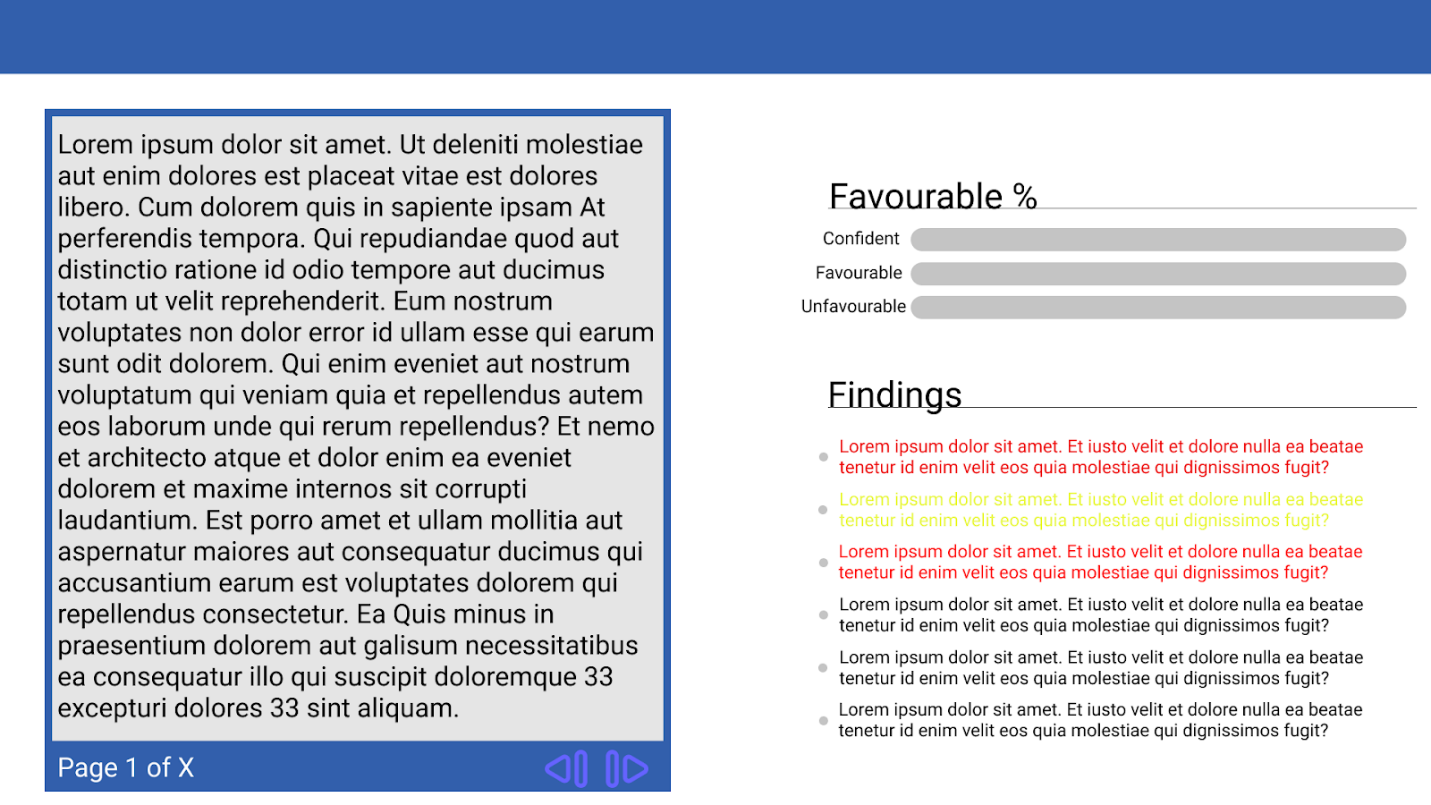


Figure : Administrator Main Page

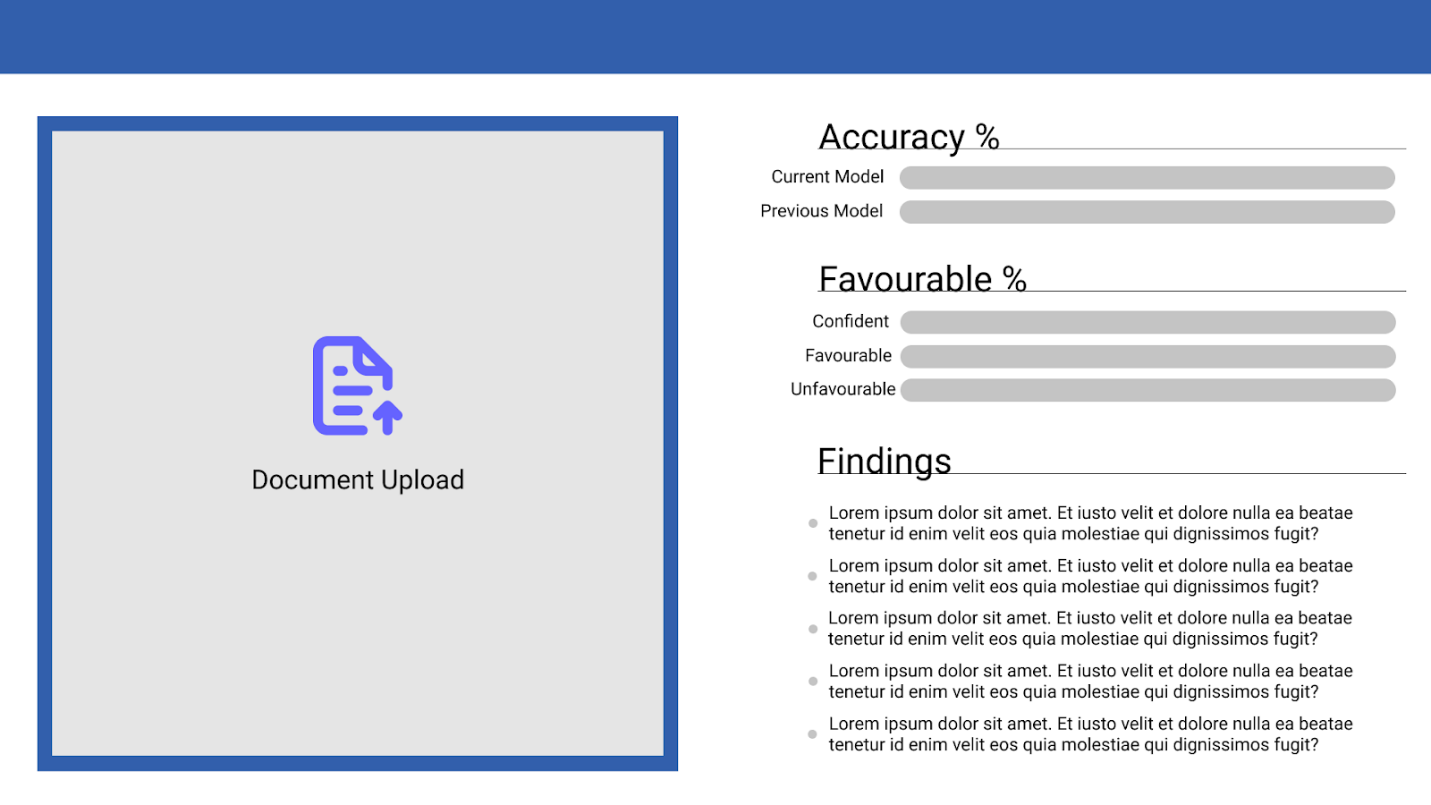
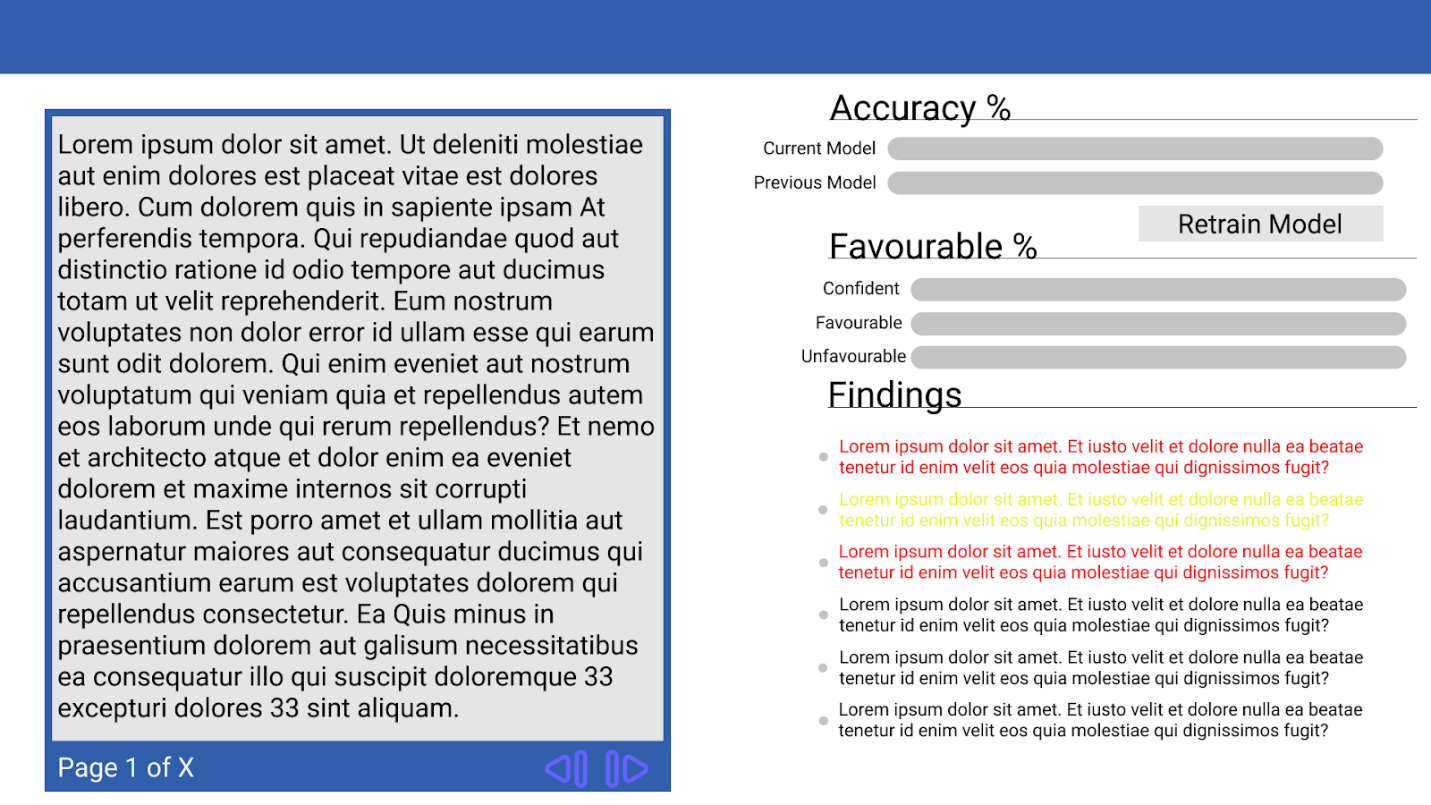


Figure : Administrator Document Results



## 5.2 Static Model – Class Diagrams

Figure : Static Model - Class Diagram

Diagram

Description automatically generated

## 5.3 Dynamic Model – Sequence Diagrams

Figure : Account Sequence Diagram

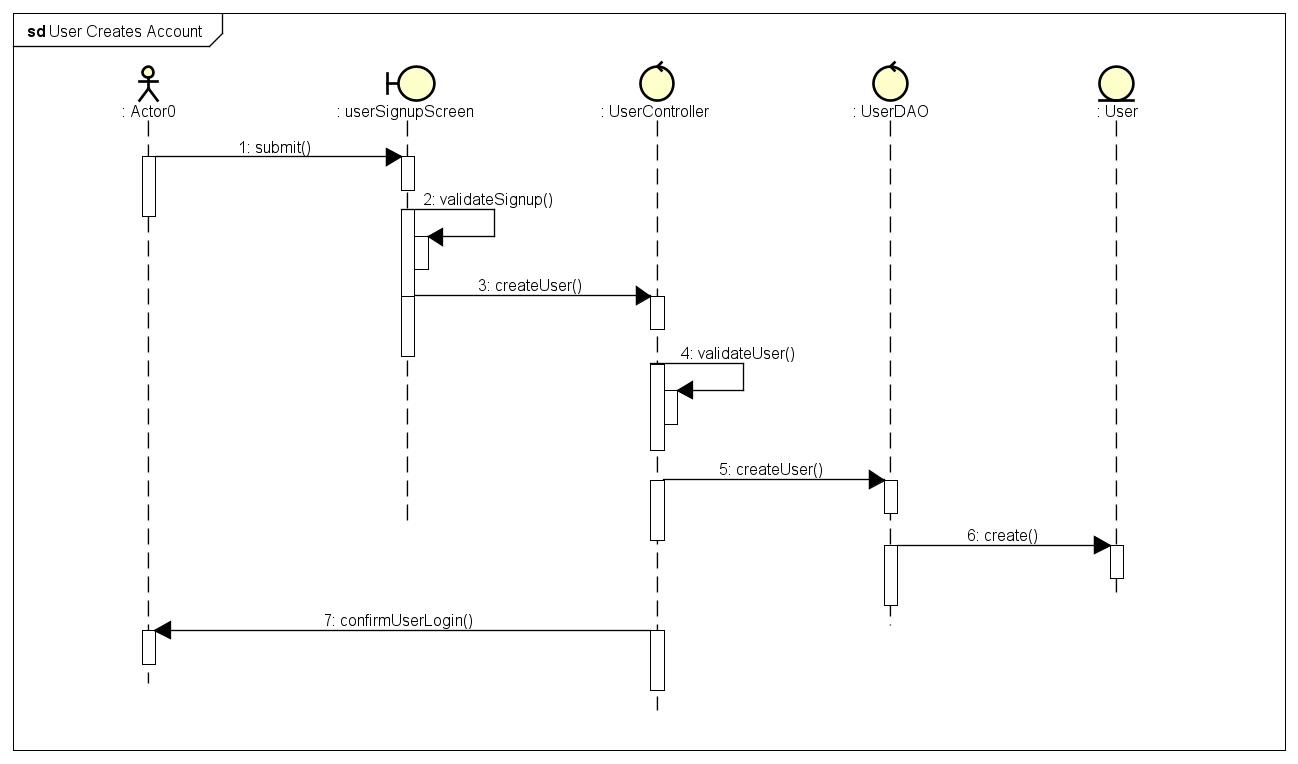
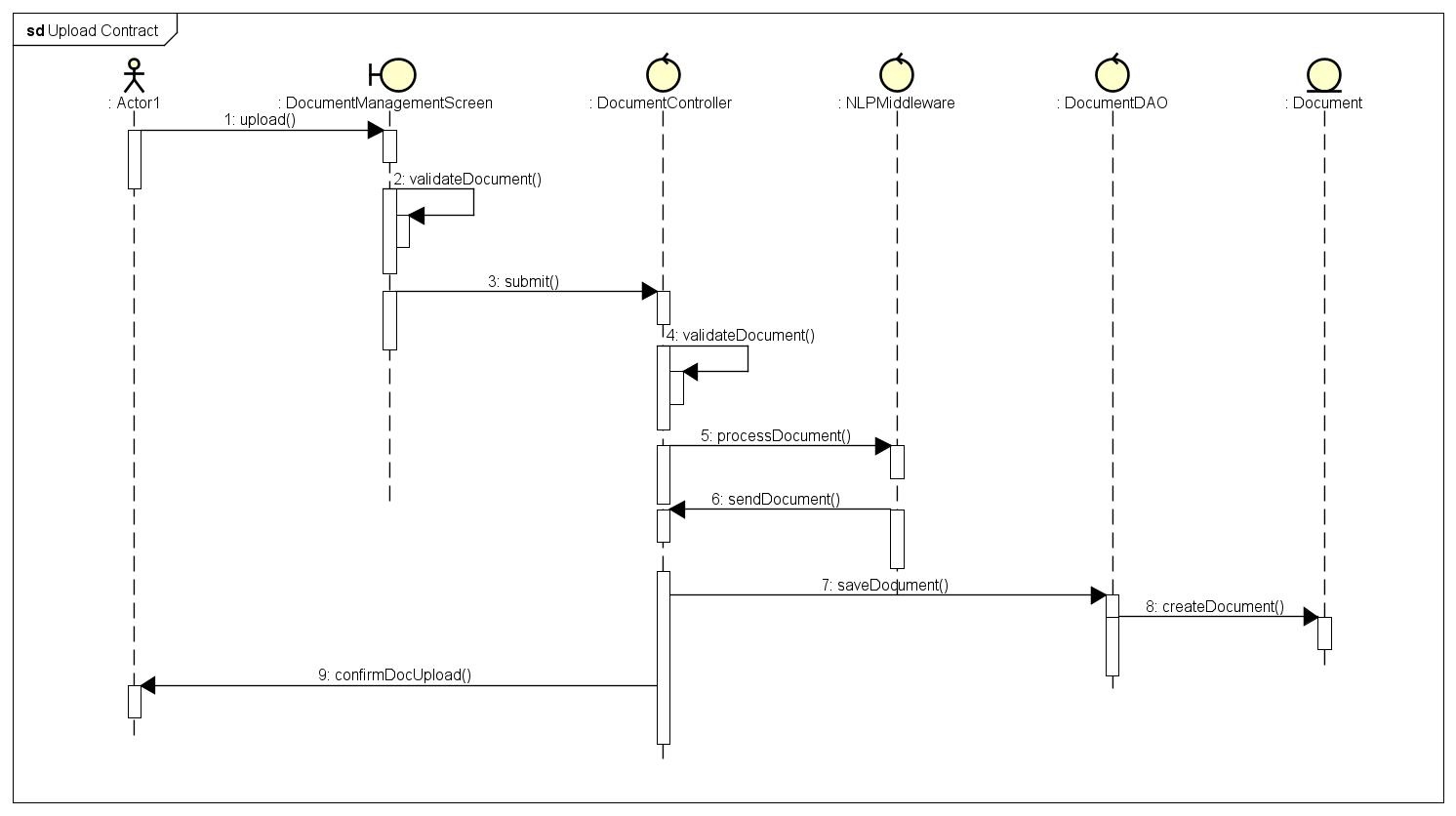


Figure : Upload Sequence Diagram



## 5.4 Rationale for Your Detailed Design Model

The user pages demonstrated in this document of the GUI can be divided into several key parts: document upload/preview, accuracy rating, favorability rating, findings. The purpose of the document upload/preview is to allow users to upload a file and show the raw text of the document from text extraction and will help users to identify whether there were issues in reading any part of the document. The accuracy rating available to administrators is to show and compare model performance between the current model and previous model. This allows for retraining the model and will lead to a new screen (not shown here) that allows for model selection and configuring the model. The favorability rating will show the model’s confidence level and keywords that are favorable and unfavorable from the document. The findings will provide the sentences in which those keywords are found to allow the user to see the context.

The class diagram shows how we intend to implement our dashboard and how the components will interact. The interconnected classes and their dependencies in this diagram cover the process from when the user logs in and uploads a document to the results presented to the user after processing the document.

The sequence diagram shows the implementation of use cases into the design’s workflow. On both diagrams we can see how the response from the user is handled in all the three layers of the system architecture: presentation, service, and data. The façade design is demonstrated by the controller class that attributes services to other classes in the application tier. The DAO design pattern is being implemented by the DAO class in the data tier to create the entity and save it in the database.

## 5.5 Traceability From Requirements to Detailed Design Model

Table : Requirements Traceability

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Req# | Description | Description | Description | #PUC | Description |
| **Functional Requirements** | F1 | User creates account | User wants to sign-up for an account | System validates user information  System creates account for user | 1 | Sign up |
| F2 | User logs in to account | User wants to log in to existing account  Ask for authentication of login credentials | System displays login page  System displays main menu from the dashboard | 2 | Login Page |
| F3 | User sets system to their preferred language | User sets system to their preferred language | System gives an option to choose from multiple languages | 3 | Setting Page |
| F4 | Upload Contract | User wants to upload desired contract in pdf file format | Display user uploaded document in the correct format | 3 | Main Page |
| F6 | Text Extraction | The system will extract text from user input documents | System cleans and processes data from 6. and prepares and organizes text in the correct format | 3 | Main Page |
| F7 | Check Contract Favorability Skew | User wants to view flagged keywords and phrases | The model  determines which keywords and phrases and provide insight on the  favorability of flagged information | 4 | Dashboard |
| F8 | View ML model performance | User shall be able to view overall performance of the ML model | The model provides information regarding words that did not meet the  threshold parameters specified by the user. | 4 | Dashboard |
|  | F9 | View Logs | System keeps logs to ensure that user is valid elevated within the system and has successfully logged in | System provides logs regarding information about application status | 2 | Login page |
| **Usability** | NF1 | The system must work on OSX, Windows, and Unix-based systems | User can use different Operating System | Display valid dashboard in OSX, Windows, and Unix environments | 4 | Dashboard |
| NF2 | The system may support multiple languages | A possible add-on would be to support contracts that may be in other languages | Display the options from F3. | 5 | Setting Page |
| NF3 | The system must provide sorting functionality | System must sort using highest confidence levels, favorable/unfavorable  keywords, etc. | Display the sorted information in the dashboard | 4 | Dashboard |
| **Availability** | NF4 | The system must retrieve contract documents one at a time | A user will be able to input one pdf document at a time instead of allowing users to  input multiple files at a time. | System displays appropriate screen for upload successful/failed status | 3 | Main Page |
| NF5 | The system should not take longer than 3 seconds to login the user | The system should validate user login by cross checking email and password and this  should be complete in less than 3 seconds | System displays appropriate screen for login  status(successful/failed) | 2 | Login Page |
| NF6 | The system must be able to process contract analysis within hours | User request to process contract | System process contract analysis within hours depending on the document size | 3 | Main Page |
| **Security** | NF7 | Passwords must be hashed and stored securely in the database | The application will not be available to all users, valid users must authenticate and verify  identity with their title/position, employee id (if applicable), and password | System stores password in secure database | 2 | Login Page |
| NF8 | Sensitive information shall be encrypted for in-flight data | Information contained within the contracts that the program works with may be  sensitive and will require some form of encryption when the application is run over a  network | System will encrypt all sensitive data in the application | 3 | Main Page |
| NF9 | System will automatically sign out within 15 minutes if no work is in progress and no response  from the user is detected | Login session will be terminated to prevent sensitive data from being leaked when users  are no longer using the application and no work is running | System will notify users session termination and ask users to log in again | 2 | Login page |
| NF10 | Passwords must contain at least 8 characters including at least one special character, upper case  letter, and number | Valid users will need to create a strong password as the program will deal with sensitive  information | Display message and notify users to reenter   strong password combination | 1 | Sign up Page |
| NF11 | Password delay after 10 consecutive failed login attempts | Too many failed login attempts will require the user to verify their identity or wait a  certain amount of time to prevent brute-force attempts | Display attempt amount left in each trial | 2 | Login Page |

# 6. Test Plan

## 6.1 Requirements/Specifications-based System Level Test Cases

Table : Login-1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | Login-1 | | |  | | |
| **Test Case Description** | | Login – Valid test cases | | |  | | |
| **Prerequisite** | | A valid user account | | |  | | |
| **Scenario Number** | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Launch application | | Open desktop app/ UI | Peraton App login page | | Pass | Launch successful |
| 2 | Enter correct Email & Password and hit login button | | Email id : test@peraton.com  Password: \*\*\*\*\*\* | Login success | | Pass | Login successful |

Table : Login-2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | Login-2 | | |  | | |
| **Test Case Description** | | Login – Invalid test case | | |  | | |
| **Prerequisite** | | N/A | | |  | | |
| **Scenario Number** | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Enter invalid Email & any Password and hit login button | | Email id : invalid@gmail.com  Password: \*\*\*\*\*\* | The email address or phone number that you've entered doesn't match any account. Sign up for an account | | Fail | Invalid login attempt stopped |
| 2 | Enter valid Email & incorrect Password and hit login button | | Email id : valid@peraton.com  Password: \*\*\*\*\*\* | The password that you've entered is incorrect. Forgot Password? | | Fail | Invalid login attempt stopped |

Table : Upload-1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | Upload-1 | | |  | | |
| **Test Case Description** | | Upload – Valid test case | | |  | | |
| **Prerequisite** | | User account with permission to upload | | |  | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Upload contract in pdf format | | Contract format .pdf | Contract successfully uploaded | | Pass | upload successful |
| 2 | Upload contract in word format | | Contract format .docx | Contract successfully uploaded | | Pass | upload successful |

Table : Upload-2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | Upload-2 | | |  | | |
| **Test Case Description** | | Upload – Negative test case | | |  | | |
| **Prerequisite** | | User account with permission to upload | | |  | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | Upload contract in excel format | | Contract Format extension .xlsx | Please enter the correct contract format. (word document or pdf format) | | Fail | upload unsuccessful |
| 2 | Upload corrupted file | | Corrupted file | Unable to read file. | | Fail | Upload unsuccessful |

Table : Contract Favorability Skew-1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | Contract Favorability Skew-1 | | |  | | |
| **Test Case Description** | | Contract Favorability Skew– Valid test case | | |  | | |
| **Prerequisite** | | Processed Data (Contract) | | |  | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | View Favorability Skew | | Processed document | Confidence Level - 80%  Favorable - 70%  Unfavorable - 30% | | Pass | Confidence level of model meets the minimum threshold |

Table : Contract Favorability Skew-2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | Contract Favorability Skew-2 | | |  | | |
| **Test Case Description** | | Contract Favorability Skew– Invalid Test case | | |  | | |
| **Prerequisite** | | Processed Data (Contract) | | |  | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | View Favorability Skew | | Processed document | Confident % - 40%  Favorable - 30%  Unfavorable - 70% | | fail | Confidence level of model does not meet the minimum threshold |

Table : View ML Model performance-1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | View ML Model performance-1 | | |  | | |
| **Test Case Description** | | View ML model performance – Valid Test case | | |  | | |
| **Prerequisite** | | Processed Data (Contract) | | |  | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | View ML model performance results | | Processed document | Most recent ML model performance compared to previously trained model | | Pass | View ML model performance successful |
| 2 | View ML model performance results | | Processed document | The words that didn’t meet the threshold parameters are: “examples..” | | Pass | View ML model performance successful |

Table : View ML Model performance-2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Description** | | View ML Model performance – Invalid Test case | | |  | | |
| **Prerequisite** | | Processed Data (Contract) | | |  | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Test Result** | **Test Comments** |
| 1 | View ML model performance with empty results | | Processed document | No available previously trained model | | fail | View ML model performance unsuccessful |
| 2 | View ML model performance with empty result | | Processed document | Couldn’t show words that didn’t meet the threshold parameters specified by user | | fail | View ML model performance unsuccessful |

## 6.2 Traceability of Test Cases to Use Cases

Table : Traceability of Test Cases to Use Cases

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Test Scenario ID** | **S.No** |
| R02 | Login-1 | 1,2 |
| R02 | Login-1A | 1,2 |
| E02 | Upload-1 | 1 |
| E02 | Upload-1A | 1 |
| E03 | Text Extraction-1 | 1 |
| E03 | Text Extraction-1A | 1 |
| Y04 | Data Processing-1 | 1 |
| Y04 | Data Processing-1A | 1 |
| Y01 | Contract Favorability Skew-1 | 1 |
| Y01 | Contract Favorability Skew-1A | 1 |
| Y02 | View ML model performance-1 | 1,2 |
| Y02 | View ML model performance-1A | 1,2 |

## 6.3 Techniques Used for Test Generation

Unit testing

The unit is the smallest and indivisible part in the software that is tested to not have analysis or programming errors. This white-box technique is considered one of the most popular in software development and is supported by many testing tools.

Statement Coverage

This white-box technique is widely used in the software testing industry. The statement coverage is given by the coverage percentage from the formula: actual number of statements covered divided by the total number of statements and the number of unreachable statements in the program. The statement coverage does not subsume decision and condition coverage.

Integration Testing

This black-box technique is performed when two or more components are combined to create a new subsystem or a system. This technique is intended to reveal any error that may exist during communications among the components being integrated. Combined with unit testing, these two become a powerful testing bundle for most software. Differently from the previous two methods, the integration testing is designed to determine whether the system meets the requirements.

Acceptance Testing

Testing conducted to determine if a system satisfies its acceptance criteria and to enable the customer to determine whether to accept the system. This method is also a black-box based testing.

The quality of the testing techniques is evaluated by the following criteria:

1. Number of test cases generated

Testing methods that the number of test cases grows exponentially is less desired than those that grow in a linear fashion.

1. Number of defects revealed

There is not a single testing technique that covers 100% of the program. Some techniques are able to cover more portions of the code than others.

1. Duration to complete testing analysis

Testing methods that take longer to complete their analysis are very time-consuming during the testing phase, e.g., mutation testing versus statement coverage.

## 6.4 Assessment of the Goodness of Your Test Suite

The metrics used to determine the goodness of our test suite is whether our program could handle the established test cases we made and whether they had as wide coverage as possible for all the key portions of our program. Below are some of the test cases and results tested for our application.

**Testing**

Test case #1: Check incorrect input format with excel file

Figure : Test 1 - File Upload Test

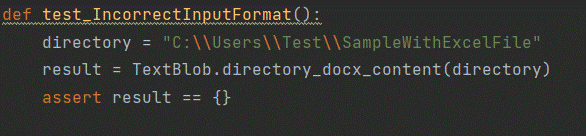
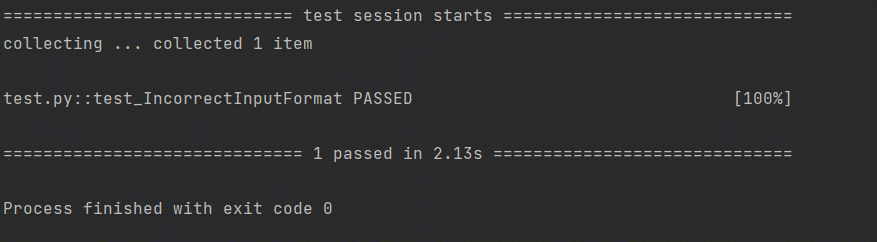
****

Figure : Test 1 - Result

****

Test case #2: Check correct input format with docx format

Figure : Test 2 - Verify Valid Upload

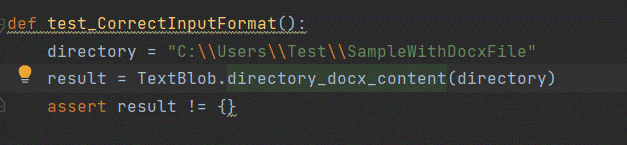
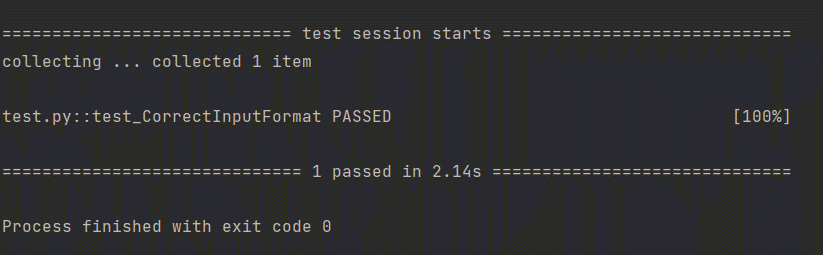
****

Figure : Test 2 - Result

****

Test case #3: Check if data is labeled correctly for positive case

Figure : Test 3 - Positive polarity test

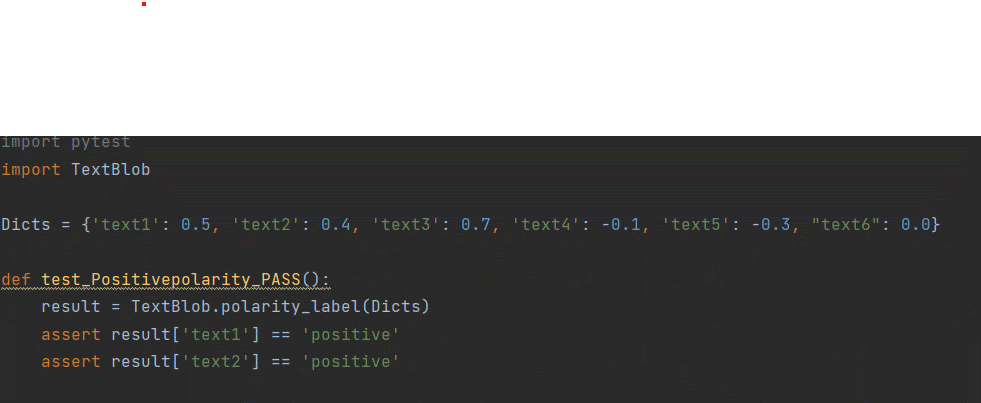
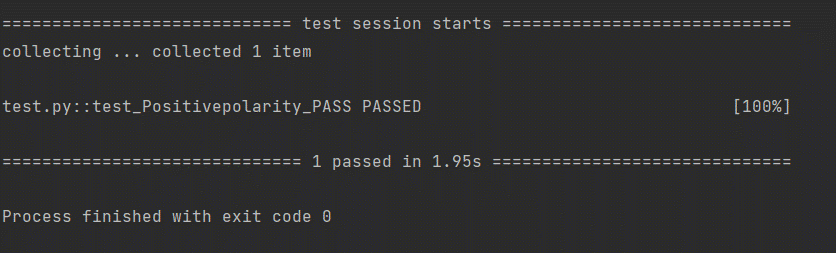


Figure : Test 3 - Result



Test case #4: Check if data is labeled correctly for non-positive case

Figure : Test 4 - Non-positive polarity test

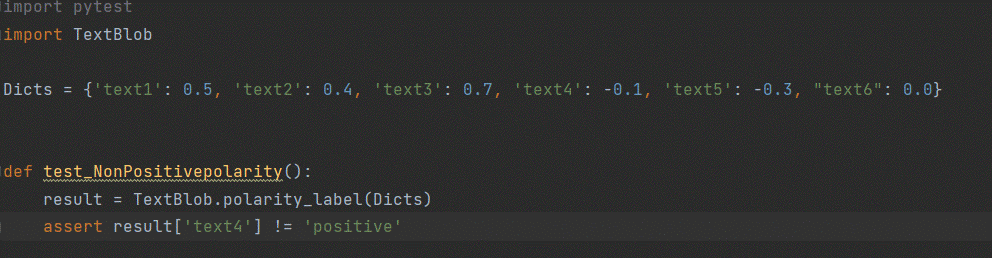
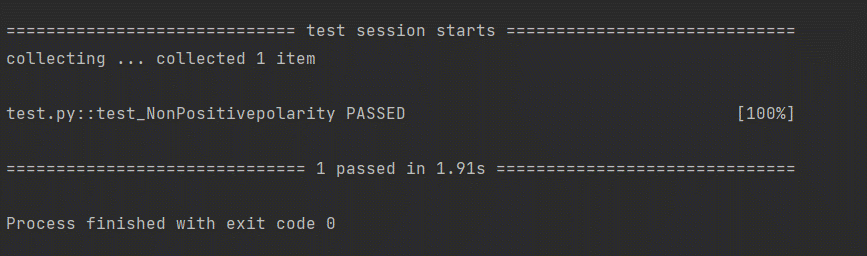


Figure : Test 4 - Result



Test Case #5: Check if data is labeled correctly for negative case

Figure : Test 5 - Negative polarity test

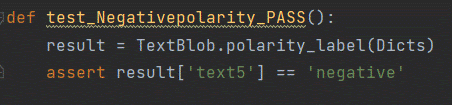
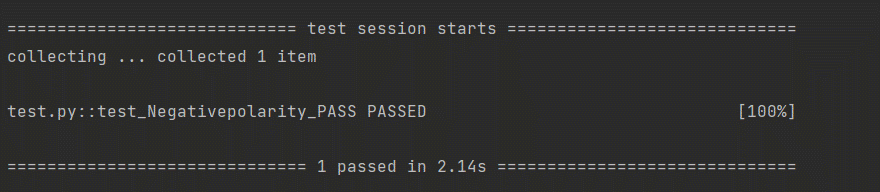


Figure : Test 5 - Result



# Acknowledgment

We thank our sponsor and mentor, Dave Gibson, for meeting and communicating with our team regularly and providing us with the opportunity to work on an interesting project that allowed all of us to learn new technologies that will further our skills. We also thank him for meeting our requests in a timely manner and answering any questions we had.

We thank our Instructor for facilitating projects between the sponsors and students allowing us to gain insight into working in a professional setting and discover new and interesting projects.

# 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].
5. “TextBlob API Reference,” API Reference - TextBlob 0.16.0 documentation. [Online]. Available: https://textblob.readthedocs.io/en/dev/api\_reference.html. [Accessed: 02-Dec-2021].
6. “Django Documentation,” Django. [Online]. Available: https://docs.djangoproject.com/en/3.2/. [Accessed: 02-Dec-2021].
7. “MongoDB Documentation,” MongoDB Documentation. [Online]. Available: https://docs.mongodb.com/. [Accessed: 02-Dec-2021].