Requirement Analysis

1.Introduction

1.1 Vision

Labeled data is a group of samples that have been tagged with one or more labels. A labeling system typically takes a set of unlabeled data and augments each piece of it with informative tags. After obtaining a labeled dataset, machine learning models can be applied to the data so that new unlabeled data can be presented to the model and a likely label can be guessed or predicted for that piece of unlabeled data.

1.2 Scope

The aim of this project is to design and implement a data labeling system in object-oriented manner. The data labeling system has multiple labeling mechanisms. These are; random, machine learning, simple search, user interface, and sentence labeling mechanism. In our program, instances of the dataset will be labeled based on the labeling mechanism provided by the user. There is an authentication mechanism for human users too.

2. Functional Requirements

- System must read the input file and store the dataset.
- System must be configured based on the configuration file.
- User should enter username and password for authentication.
- System must behave like bot user if username and password entered blank.
- Human user can assign labels manually.
- System must inform human user about current instance to be labeled , labels that he/she can assign and maximum number of labels.
- User should be able to assign one or more labels to an instance.
- Users should be able to label previously labeled instances according to given probability.
- User should be able to assign labels to multiple instances.
- One or more user can assign different labels to the same instance.
- The system must choose most frequent label as a final label for each instance.
- The system must calculate 3 different performance metrics (User, Instance, Dataset) after each assignment.

- System must print its actions to the command line and the log file.
- System must store the assignments in the dataset.
- The system must print performance metrics and outputs which contains assignments done so far when it is terminated suddenly.
- System must print the labeled dataset to an output file.
- The system must be able to restart from where it terminated.

3. Nonfunctional Requirements

3.1 Usability

- Outputs, performance metrics and logs of the system should be printed with an organized manner to be easily understood by the user
- The system should be a multi user system.
- The system should be a multi-dataset system.

3.2 Flexibility:

- The system should support easily pluggable labeling mechanisms.
- The system should be able to integrated to a user interface with minimal changes.
- The system should support different labeling mechanisms.
- The system should support scenarios in which words/terms in a document can also be labeled.

3.3 Performance:

- The system must give an output at a reasonable time.
- **3.4 Reliability:**
- Components of the project code will be tested alongside the implementation phase to ensure that they are functional.
- **3.5 Stability:**
- System should be stable in a sense that it should work with different inputs.
- **3.6 Supportability:**
- The application must not be platform dependent, i.e., it should be able to run on any platform.

4. Glossary

Instance: An example of a particular type

Label: A word or phrase indicating that the instance belongs to a particular category or class
Final Label: Most frequently used class label for an instance.
Dataset: A group of instances
User: Someone who interacts with the system
Labeling Mechanism: A system to assign labels to the instances
Assignment: the allocation of some labels belonging to a particular instance or group of instances
User Performance Metrics: A report contains user's performance measurements based on their activity on datasets.
Instance Performance Metrics: A report contains instance's performance measurements based on their statistics.
Dataset Performance Metrics: A report contains dataset's performance measurements based on current user activity and assignments.
User Interface Labeling Mechanism: A mechanism that human users can assign labels to instances from console.
Simple Search Labeling Mechanism: It is a labeling mechanism based on searching the label texts in instances.
Separate Sentence Labeling Mechanism: This mechanism forces user to assign labels to every separate sentence in an instance and then mechanism choose a label for the whole instance based on the user's separate choices. This helps users to make more accurate assignments by reducing complexity on the side of the user.
Machine Learning Labeling Mechanism: Mechanism to label instances using Natural Language Processing(NLP) data.

5.Stakeholders:
Customers:
Murat Can Ganiz
Lokman Altın
Developers:
Eymen Topçuoğlu
Berkay Deniz
Muhammed Raşit Ayaz
Ahmet Emirhan Bakkal
Yunus Yıldırım
Vahap Gözenelioğlu
Ahmet Faruk Yılmaz
Ubeydullah Günay

6.Use Cases

Use Case: Simulation

Actors: User, Data Labeling System

Precondition: User must provide input files (config.json, dataset.json, machine learning data)

1) User starts the system.
2) System selects the dataset which determined by config.json.
3) System parses dataset.json and constructs the dataset.
4) System asks for user name and password.
5) User leaves user name and password blank.
6) System determines the corresponding data labeling mechanism based on the user type.
7) Bots start labeling instances one by one.
8) System outputs the labeled dataset to output.json.
9) System calculates and outputs performance metrics to metrics.json.

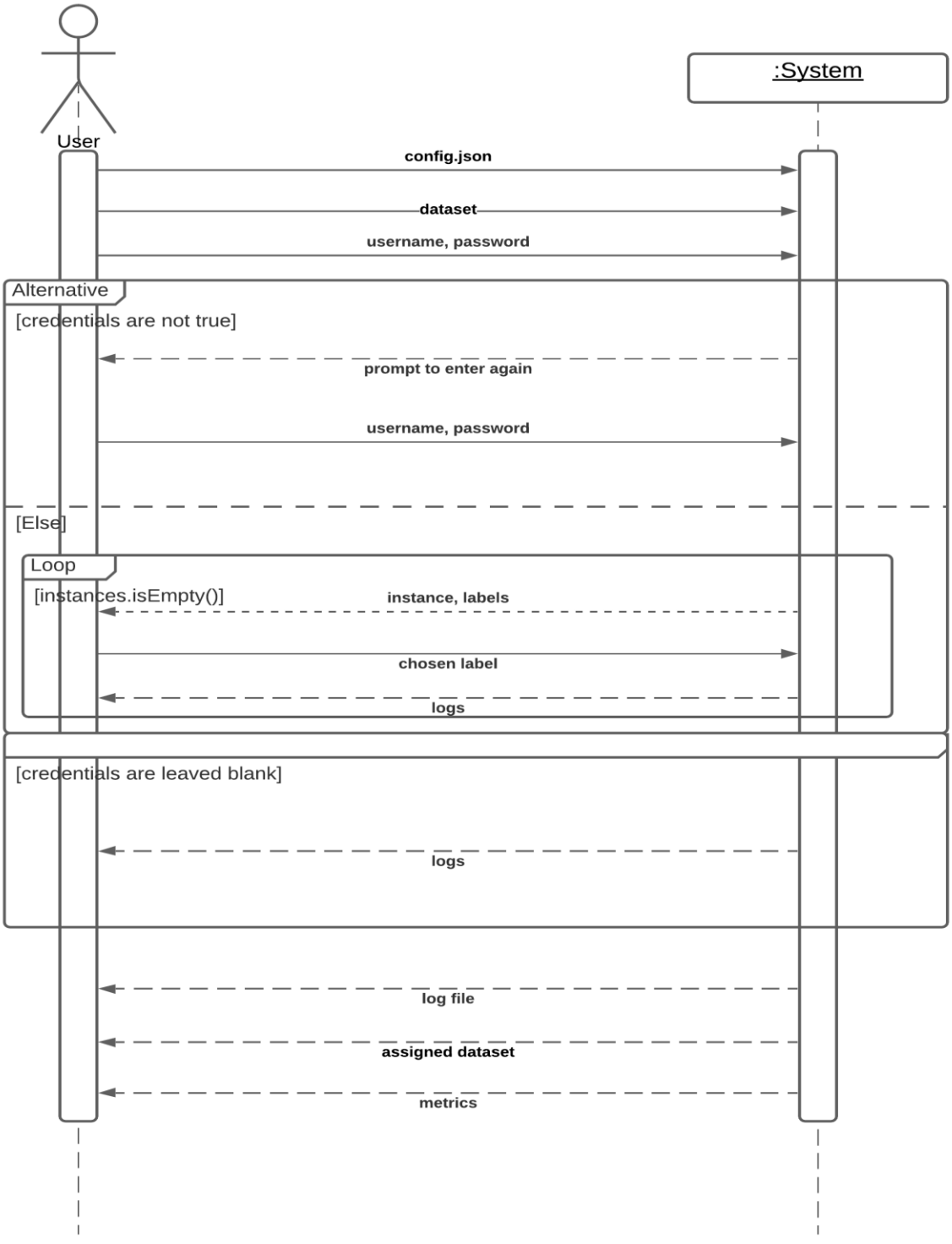
Use Case: User Interface Labeling

Actors: User, Data Labeling System

Precondition: User must provide input files (config.json, dataset.json)

1) User starts the system.
2) System selects the dataset which determined by config.json.
3) System parses dataset.json and constructs the dataset.
4) System asks for user name and password.
5) If user name and password do not match any credentials in config.json system should prompt the user to enter again.
6) System determines the corresponding data labeling mechanism based on the user type.
7) System outputs the labeled dataset to output.json.
8) System outputs performance metrics to metrics.json.

7.System Sequence Diagram



8. Domain Model