Requirement Analysis Document (RAD)

Software Requirement Specifications v1.1

Lead Software Engineers

Ahmet Hakan ŞİMŞEK

Ali Anıl REYHAN

Fatih Emin ÖGE

Hasancan ÖZEN

İsmail Ertan SARIHAN

Merve ŞAHİN

Mustafa KİBAROĞLU

Serkan EROL

Tugay SARICI

Yusuf Talha ERDEM

Customers

Murat Can GANİZ

Lokman ALTIN

INTRODUCTION

This document details the project plan for the development of “LabeIt”, our labeling mechanism program.

It is intended for designers, developers, and testers working on “LabeIt” as well as project investors. This plan will include a summary of:

- How the system will function
- Added features since last iteration
- The scope of the project from the development viewpoint
- The technology used to develop the project, and
- The metrics used to determine the project’s progress
- Overall Description

1.1 Purpose

Our project is a data labeling project. Data labeling is the process of assigning one of the several predetermined labels to a group of instances via a user interface by human experts. Instances to be labeled can be a single word or a phrase. An instance can be set to take only one label or it can take multiple labels.

1.2 Description

Program will randomly label instances in the second iteration. A reporting functionality is added to keep track of user performance and labeling operation for a particular dataset. Statistics for user are collected and compared in the context of a particular dataset or globally, and metrics for instances that are labeled by many users are calculated. The resulting reports gave us an idea about the quality of the data labeling and the performance of the users. Databases are not used in this project. We took the input data from the json file and we stored our output data in json file. While doing the labeling process, we paid attention to the maximum number of labels the instances can take. We made our project according to object oriented manner. Java language is used for programming in the project.

1.3 Glossary

Label : Usually one word (or sometimes a couple of words) that is used to tag instances in order to easily separate the data or group similar data together.

Instance : Data that consists of a word, a sentence or a phrase that needs to labeled

Labeling : The process of matching instances with labels.

Random Labeling : A labeling mechanism that handles all the labeling process itself without further user interaction. However does it so, randomly.

User : A person that labels instances by choosing one of the labeling methods (There is only random labeling for this iteration)

Dataset : A file formatted as .json that holds the necessary information

Input Dataset : A file provided by the user that holds information such as instances and labels

Output Dataset : A file created by our program that holds matched data of instances and label(s) along with user and time information

Config Dataset : A file that holds information of users and datasets. It also will include the id of the dataset that will be labeled next (a.k.a. current dataset). Config file should be easy to modify.

Consistency Probability : This means the probability to show this user a previously labeled instance again.

Consistency Percentage : The percentage of recurring instances that are labeled with same classes.

Completeness Percentage : The percentage of labeled instances in the dataset.

Entropy : Calculation of the randomness and irregularity of the random labeling mechanism.

REQUIREMENTS

2.1 Functional Requirements

- Input : User should be able to upload/insert data files which includes un-labeled instances and label sets.
- Config : Config file will be used to manage and store user and dataset information.
    - Creating User(s) : A user of the program will be able to add/create one or more users in config file by typing in their name and ID.
    - Adding Dataset(s) : Adding input datasets to the config file by providing an ID, name, and file path.
    - Choosing Next Dataset : User will be able to choose next dataset to work on by assinging it's ID as current dataset ID in config file.
    - Assingning User(s) : User will be able to assing any number of existing users in the config file to work on a particular dataset.

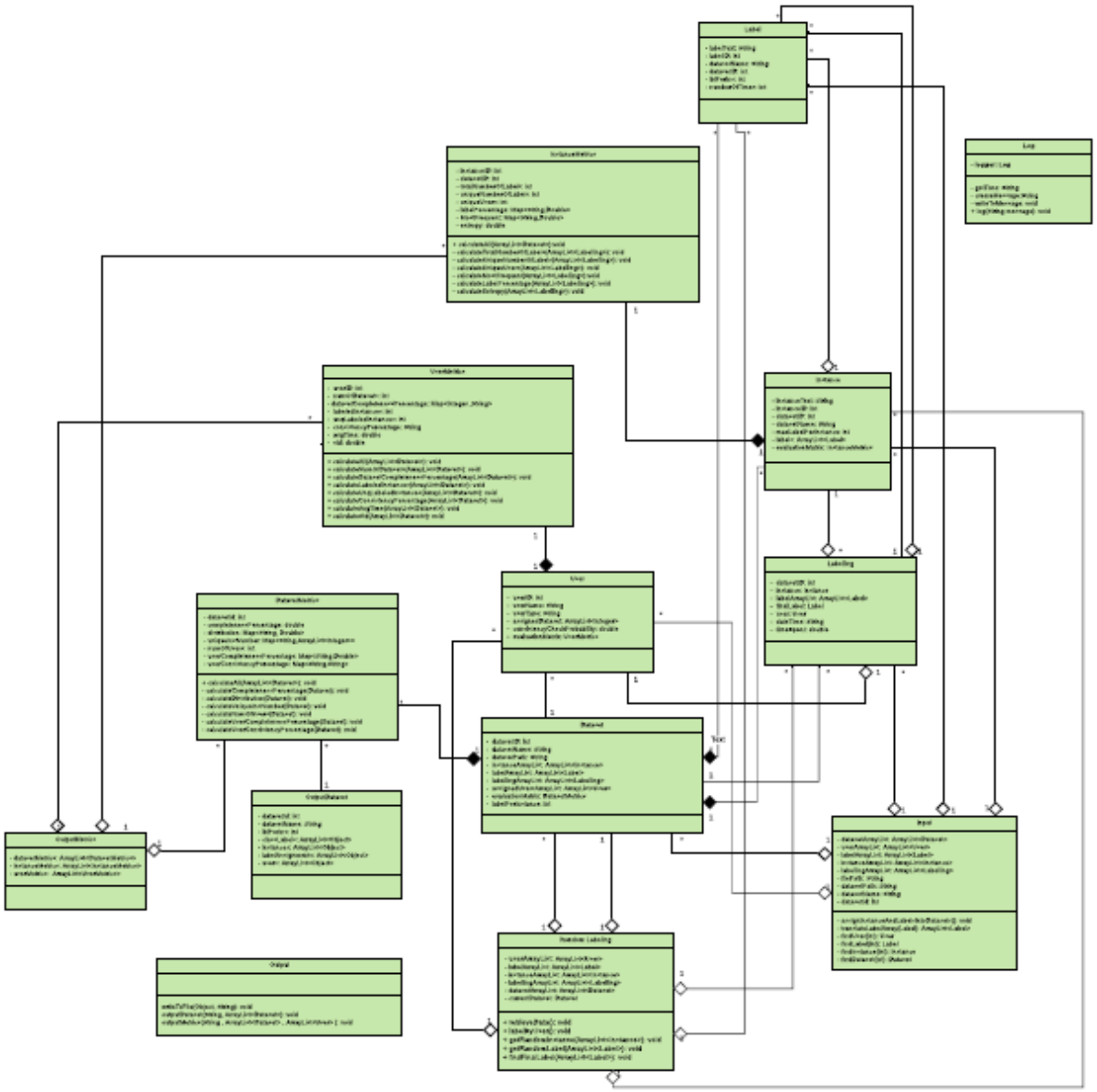
- Labeling : Users will be able to match instances with labels by using the program.
- Output : Writes the matched instance and label to the output file for later use.
- Performance Metrics : A file that holds performance information of users, instances and datasets.
    - User Performance Metrics and Reports : Logging user performance as report and as scalar metrics.
    - Instance Performance Metrics : Logging scalar metrics of instance performance
    - Dataset Performance Metrics : Logging scalar metrics of dataset performance.

2.2 Non-Functional Requirements

- Random : At this point of project, the program matches intances and labels randomly.
- LinkedList : Data such as instances, labels, user(s) or matched data are held in linked lists.
- .json Handling : json.simple and jackson libraries are used to handle .json files.
- Consistency Probability : The probability of showing current user a previously labeled instance again to.
- Performance Metrics to Calculate :
    - User Performance Metrics :
        - Number of datasets assigned
        - List of all datasets with their completeness percentage
        - Total number of instances labeled
        - Total number of unique instances labeled
        - Consistency percentage
        - Average time spent in labeling an instance in seconds

        - Std. dev. of time spent in labeling an instance in seconds
    - Instance Performance Metrics:
        - Total number of label assignments
        - Number of unique label assignments
        - Number of unique users
        - Most frequent class label and percentage
        - List class labels and percentages
        - Entropy
    - Dataset Performance Metrics
        - Completeness percentage
        - Class distribution based on final instance labels
        - List number of unique instances for each class label
        - Number of users assigned to this dataset
        - List of users assigned and their completeness percentage
        - List of users assigned and their consistency percentage

Domain Model



System Sequence Diagram