Requirement Analysis Design

Description

The goal of this project is creating a user based hand-labeling system for customers' comments. In our system, users will read the comments, which can have a single or multiple labels, of each customer and label them with appropriate predefined labels. We will create Java based object oriented models to easily maintain the process and satisfy the requirements.

Glossary

- **Artifact:** Process that support software development
- **Dataset:** Collection of useful information to manipulate
- **Design Class Diagram:** Design of a static structure that describes system's technical classes
- **Design Sequence Diagram:** Design that illustrates input and output events of the system
- **Functional Requirement:** A requirement that the system must be able to do
- **Instance:** A single occurrence of any object in Object Oriented Programming (a piece of data that is provided from a dataset , it contains an instance ID and instance text)
- **Java:** A programming language developed by Sun MicroSystems
- **Label Assignment:** Process that tags an instance with related label
- **Label:** A word that specifies or classifies the thing
- **Labeling Mechanism:** A set of processes that a user will associate an instance with given labels.
- **Non Functional Requirement :** A requirement that specifies how the system should do it.
- **Object Oriented Design:** A software development technique that describes solution in terms of objects and the operations of them
- **Output:** The client specified labeled dataset after the system run
- **Pair Programming:** A software development approach where one programmer writes code while another one watches. After some time, they swap their roles.
- **Scrum:** An iterative software development method for managing projects that describes a set of meetings and roles to help team and manage their work
- **UML:** Stands for 'Unified Modelling Language'. A diagram for developers to communicate in technical terms and software relations

List of Functional and Nonfunctional Requirements

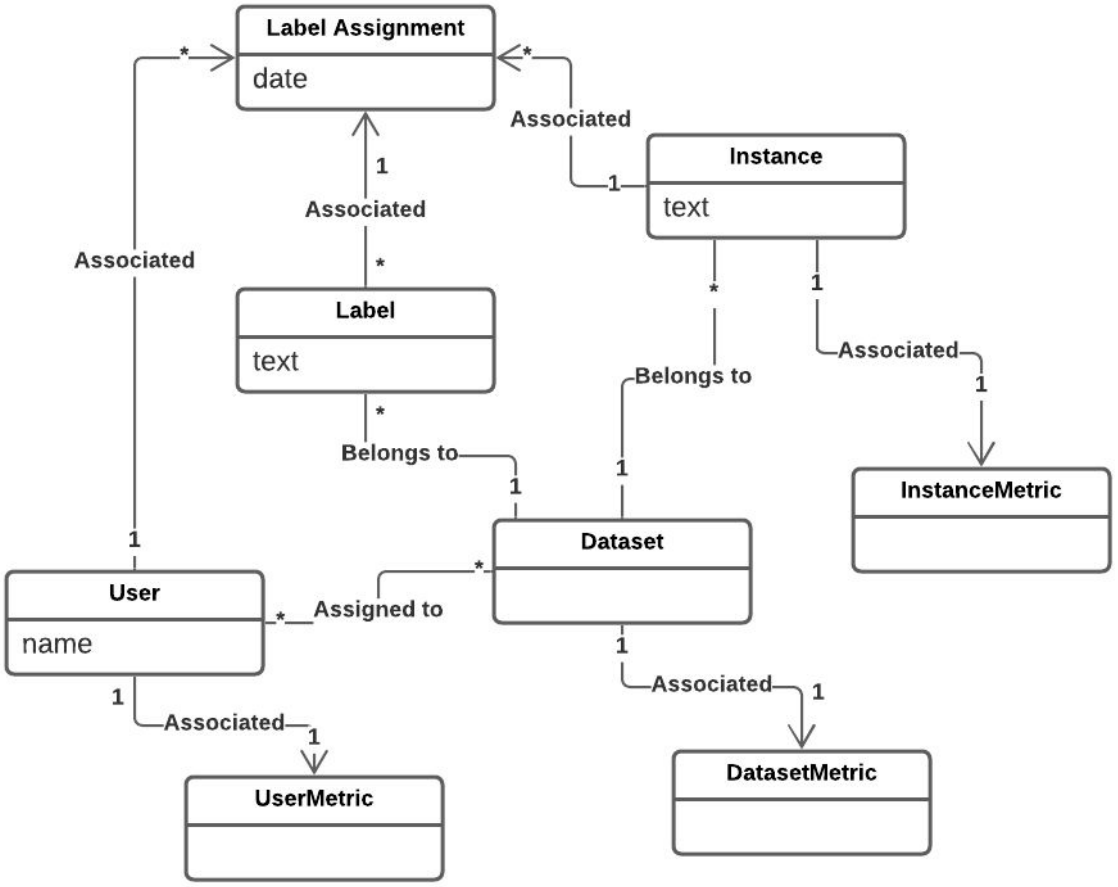
Functional Requirements

- The labeling system has to randomly assign one or more label(s) to each instance.
- The system is a multi-user system.
- Each label may be assigned to multiple instances.
- Each user can make label assignments for any instance.
- Each user can label recurrent instances multiple times.
- System should produce an output file after each label assignment for the current dataset.
- System logs every process with an informative message.
- System should be able to parse multiple datasets and create related labels.
- System calculates and reports required performance metrics and keeps them updated in each iteration.
- All users have a specified consistency check probability in the beginning.
- Each instance should be labeled depending on the most frequent class they assigned.
- System should be able to stop at any time and the customer could access the reports.

Non Functional Requirements

- The program will be run from the console.
- The system supports multi-platforms.
- System’s output should be the same dataset with its assigned labels.
- The program supports multi user and multi dataset.
- The Metric file should contain information about all users, datasets and instances in a single file.

Domain Model:



System Sequence Diagram: