Table of Contents

Introduction

Crowdsourcing is a practice of obtaining relevant information or inputs for a task or project by enlisting services of a large number of people usually through the Internet. CrowdLabeling is a type of crowdsourcing technique where multiple-choice questions are provided to the crowd to obtain true labels.

CrowdTagger is a web tool site that was built to help with collaborative labeling and its goal is to provide a platform for labelers[1] around the world to share the labeling tasks. The users of the site are assigned tasks with posts[2] and labels[3]. Each task has a description and labelers have to choose labels for different posts of different tasks. Their work will be evaluated by an algorithm using Kappa Evaluation[5] and Krippendorff's alpha[6]. The overall score of all labelings for a single task based on the posts in that task will be reported on the site. Apart from labeling, the application provides functionality to create classifiers such as SVM, random forest, train using the labeling tasks and predict the result for a new data set.

The project report demonstrates the architecture of the application followed by the functional flows and technical details. It primarily focuses on task labeling features of the application.

System Architecture

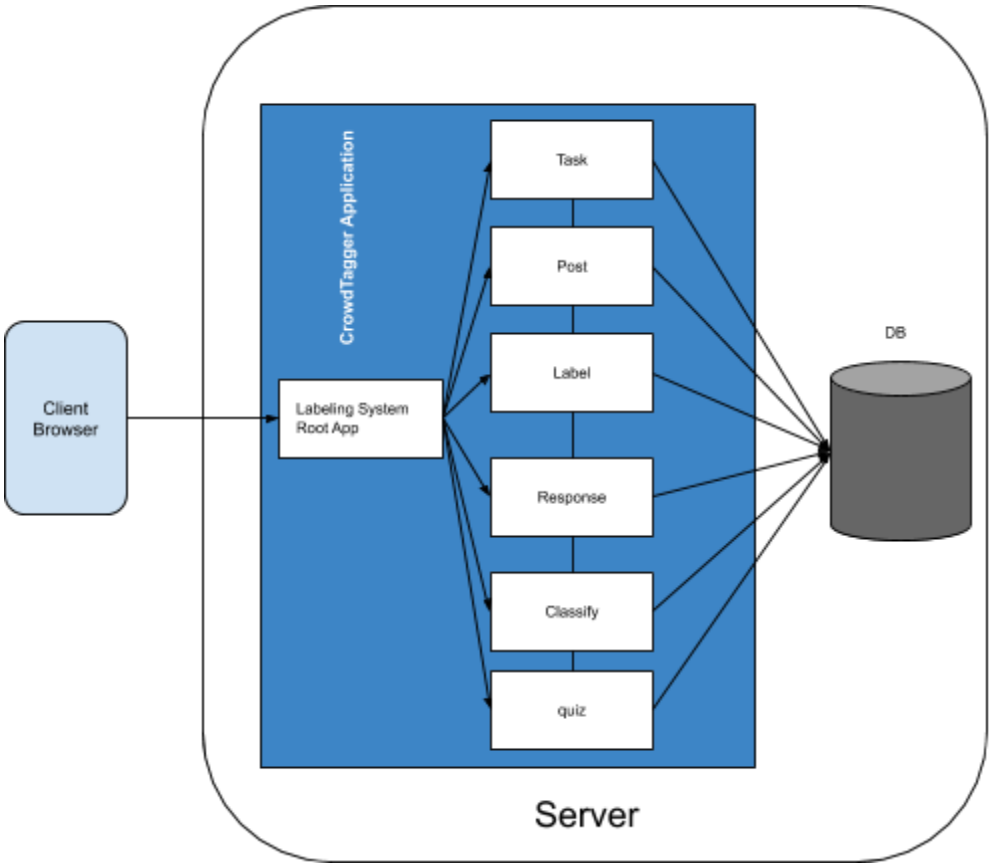


Fig 1.0: CrowdTagger Architecture

Component	**Description**
Labeling System Root App	Root component, responsible for maintaining entire application settings, configurations
Task	Manages task component related navigation URLs, views and templates
Post	Manages post component related navigation URLs, views and templates

Label	Manages label component related navigation URLs, views and templates
Response	Manages response component related navigation URLs, views and templates
Classify	Manages classify component related navigation URLs, views and templates
Quiz[4]	Manages quiz component related navigation URLs, views and templates

Table 1.0: Primary Application Components

CrowdTagger application is technically implemented using the Django[7] web framework serving on an apache tomcat server. The application is developed as several independent components holding the responsibility to perform their own scope oriented operations. The components along with their responsibilities are given in Table 1.0

Use Case Diagram

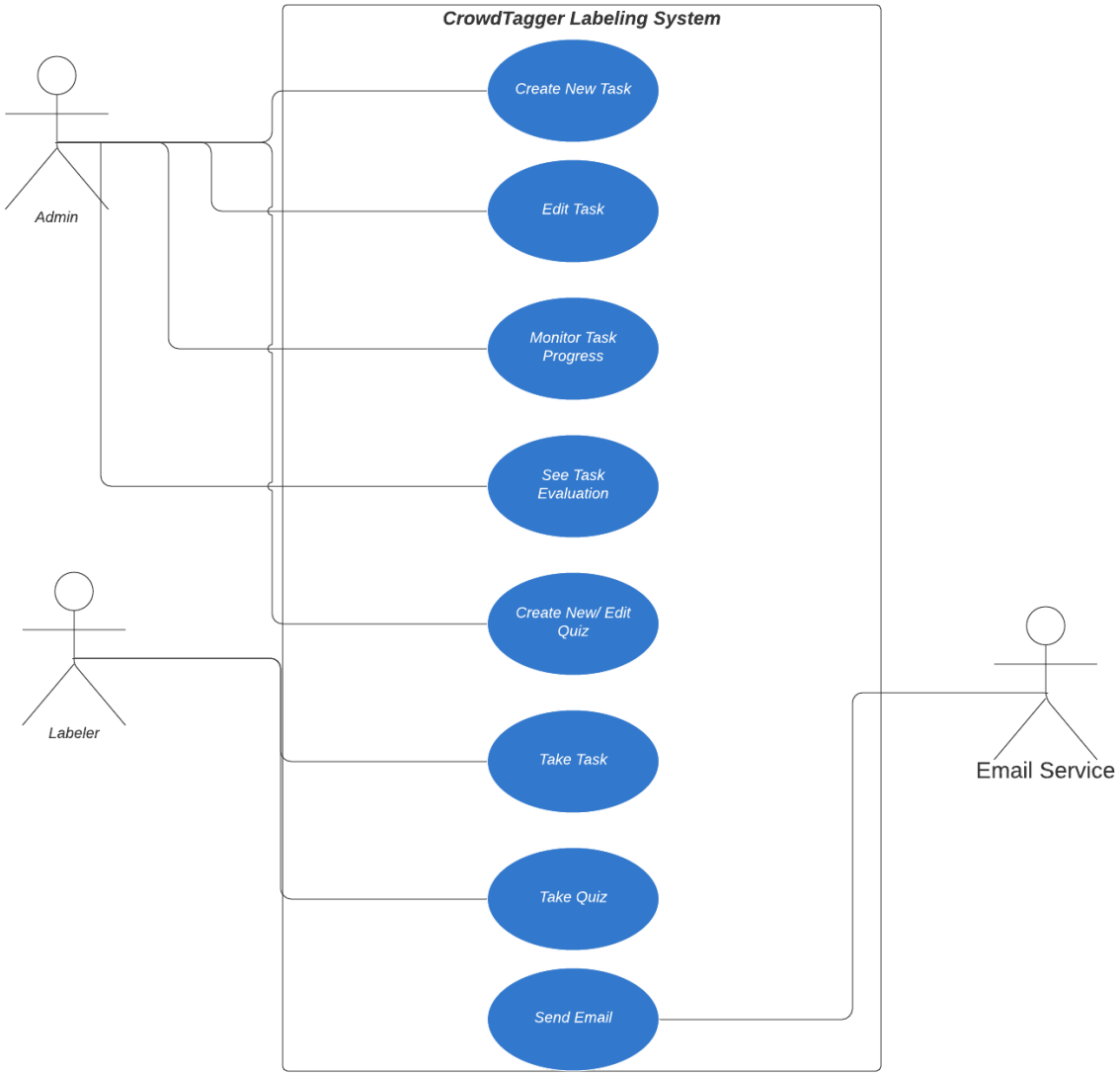


Fig 1.1: CrowdTagger Use Case Diagram[8]

CrowdTagger System has two roles assigned to users: Administrator and Labeler as shown in Fig 1.1

Administrator:

- Admins monitor the labelers' progress and observe the statistics

- Responsible
    - Create and Edit Tasks
    - Create and Edit Quizzes for the Tasks
    - Add Labelers to the Tasks
    - Monitor the Labelers' Progress and Task Evaluation

Labeler:

- Labelers choose labels for each post in the tasks assigned and take a quiz if it is a task prerequisite
- Responsible
    - Labeling Post
    - Taking Quiz

Data Model

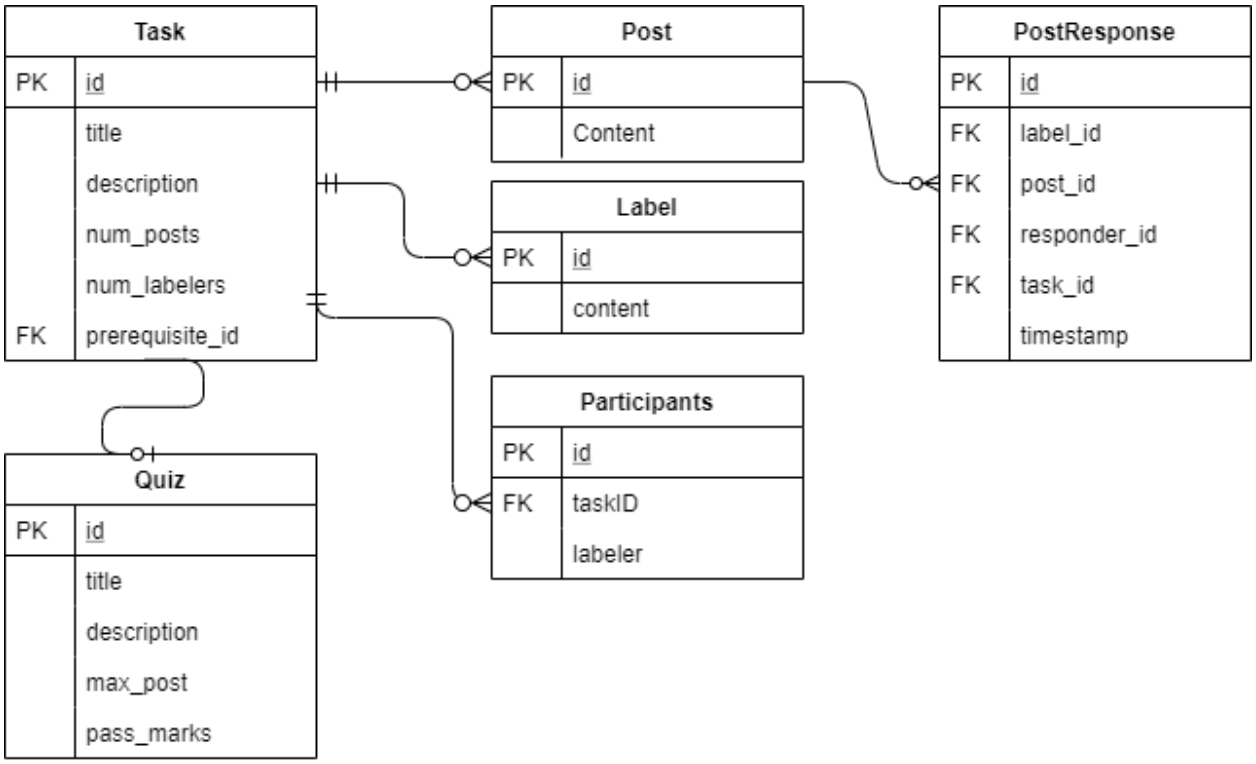


Fig 1.2: Task Labeling ER Design[10]

CrowdTagger task labeling data model comprises the following tables as shown in Fig 1.2.

1. **Task:** Maintains details about the task. On creation of a task, the details are inserted into this table.
2. **Post:** Maintains details about the posts associated with the task. On task creation, the associated posts are inserted into this table.
3. **Label:** Maintains details about labels associated with tasks. On task creation, the associated labels are inserted into this table.
4. **Participant:** Maintains labeler details associated with the task. On task creation, the associated labelers are inserted into this table.

5. **Quiz:** Maintains details associated with a quiz.
6. **PostResponse:** Maintains labelers post responses. Once the user responds to a post a postResponse record is inserted.

Associativity

1. Each task can have
    a. 1 or more posts associated with it
    b. 1 or more labels associated with it
    c. 1 or more labelers/participants associated with it
    d. 0 or 1 quiz, depending on the prerequisite
2. Each Post has 0 or more postReponses

Administrator Flows

New Task

"Create new task" functionality provides a way for the admin to create a new task for labeling. The following section provides the step to create a task

Task Creation Steps:

- On the home page, as shown in Fig 1.4 click on Create New Task, this navigates to create a new task page as shown in Fig 1.5
- Enter the title of the task, e.g., "my-task".
- Enter the task description(optional).
- Enter min number of posts, your posts should contain at least this number of rows.

- Enter min number of labelers, you should at least assign the task to this number of labelers.
- Load post and label data.
    - You can choose to load post/label from a CSV file(in this sample format)
    - Load from an existing database
    - You can view your data immediately after you upload your CSV file or choose your post table.
- Choose the quiz for this task
- Enter the labelers to participate in the task, separated by a line.
- Click submit to create the task

Technical Details:

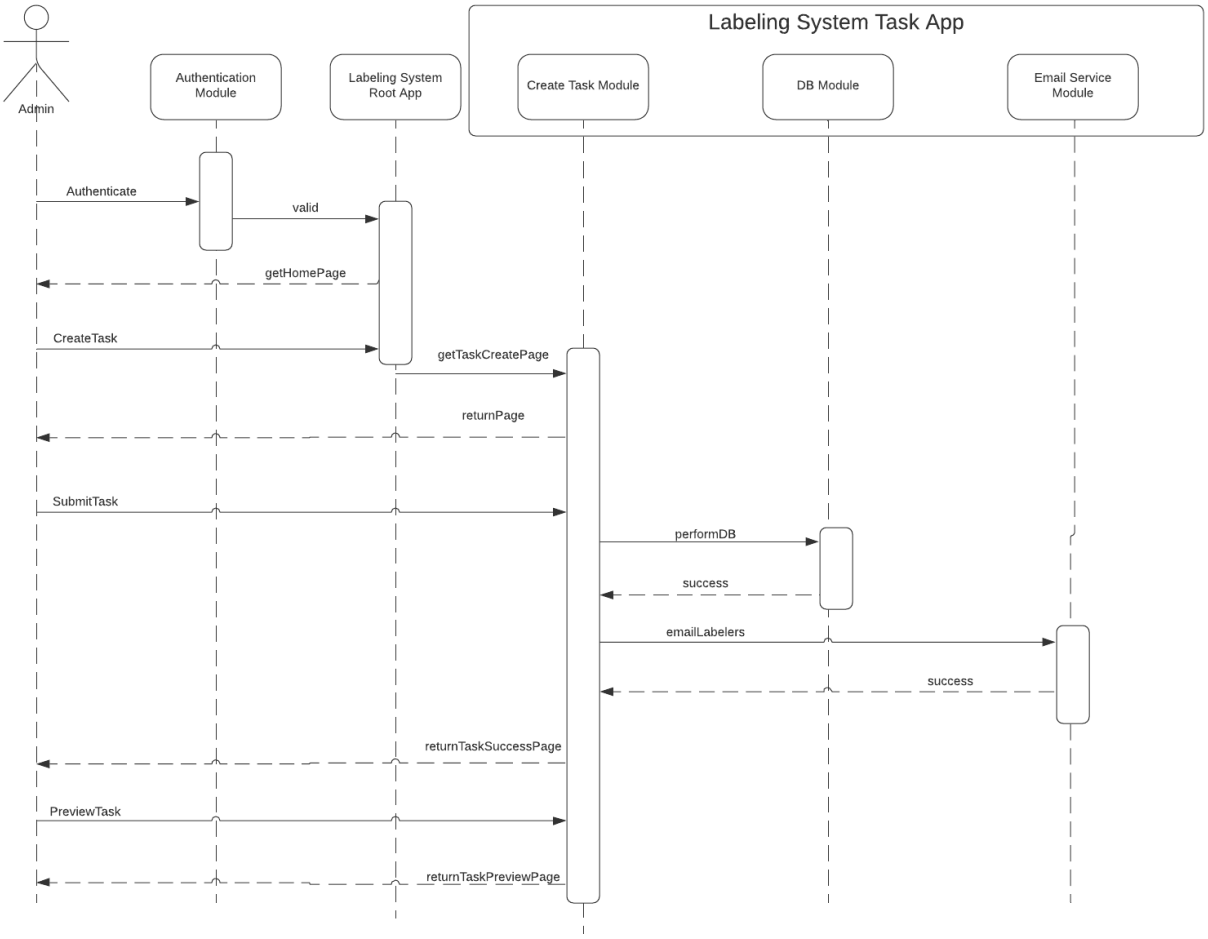


Fig 1.3 Sequence Diagram[9] - Create New Task

As shown in the above Fig 1.3, on submit, the create task module performs the following:

- Reads the submitted form and captures task fields
- Captures posts, labels either by parsing the input CSV/connecting to the database based on the selected option by the user
- Performs DB operations i.e inserts task fields into task table, inserts associated posts, labels in posts and label table respectively, and inserts labelers information into participants table
- Send an email to the labelers regarding the created task

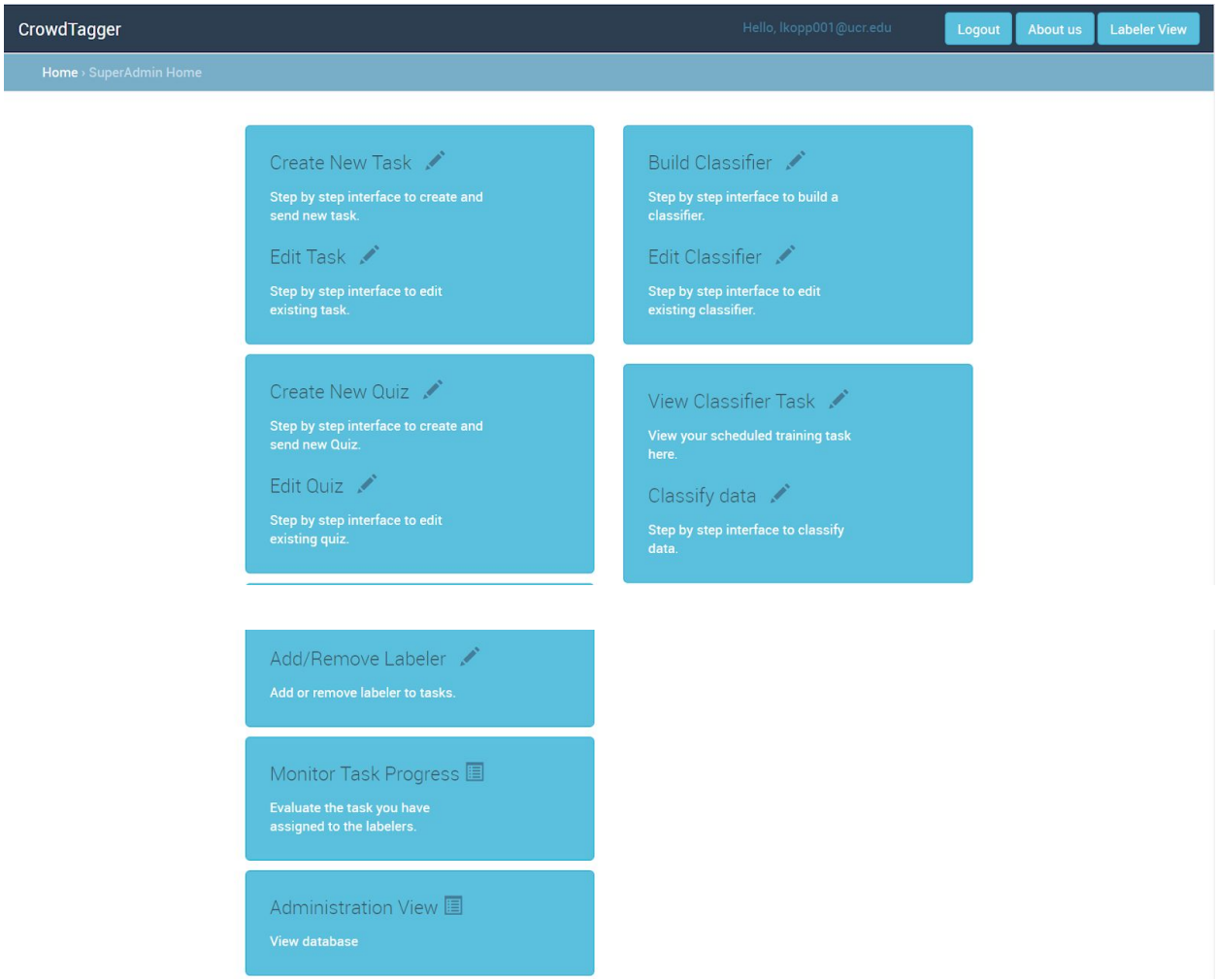


Fig 1.4: Admin Home Page

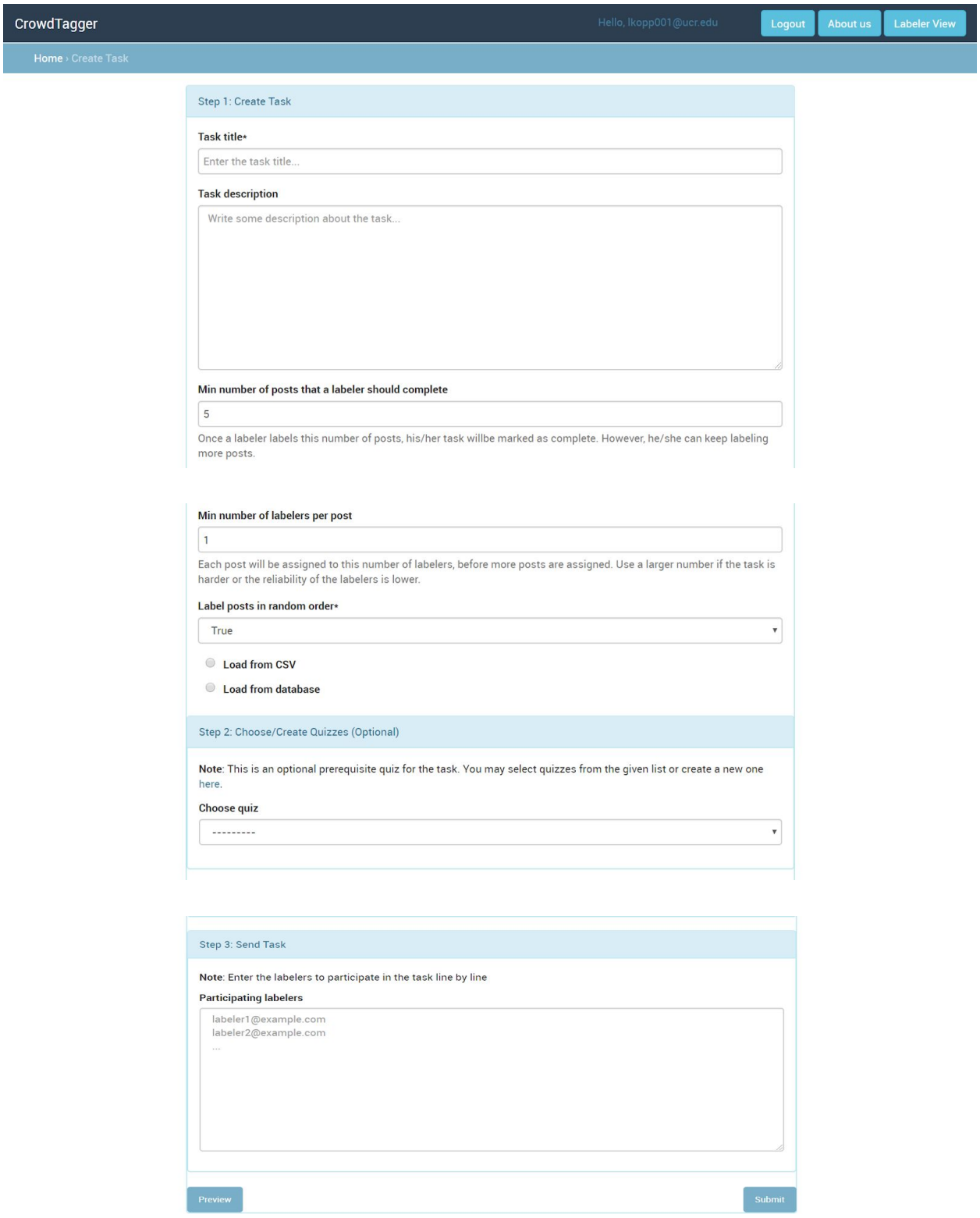


Fig1.5: "Create New Task" Page

Monitor Task Progress and Evaluate a Task

An important function as an admin is to monitor the task progress and evaluate a task. This feature provides a way to perform task monitoring and evaluation

Task Monitoring and Evaluating Steps:

- On the home page, as shown in Fig 1.4 click on Monitor Task Progress, this navigates to task evaluation list page as shown in Fig 1.7
- The task evaluation list page shows all the tasks associated with the user along with assigned labelers and responded labelers. Also, this page provides information about no of responded tasks per labeler
- Click on the “see evaluation” button to view evaluations of a particular task. On click, the user is redirected to the task evaluation detail page as shown in Fig 1.8
- The task evaluation detail page provides detail statistics on inter rate reliability of the labeling task
    - The first table displays Kappa Evaluation of all the labelers that finish the task. The pairwise kappa coefficient and total Krippendorff's alpha or download the result as a CSV file in the name of kappa-${tasktitle}-${taskid}.csv can be viewed
    - The second table displays details about the responses of all the labelers that finish the task. The response of each labeler and the majority vote or download the result as a CSV file in the name of eval-${tasktitle}-${taskid}.csv can be viewed

Technical Details:

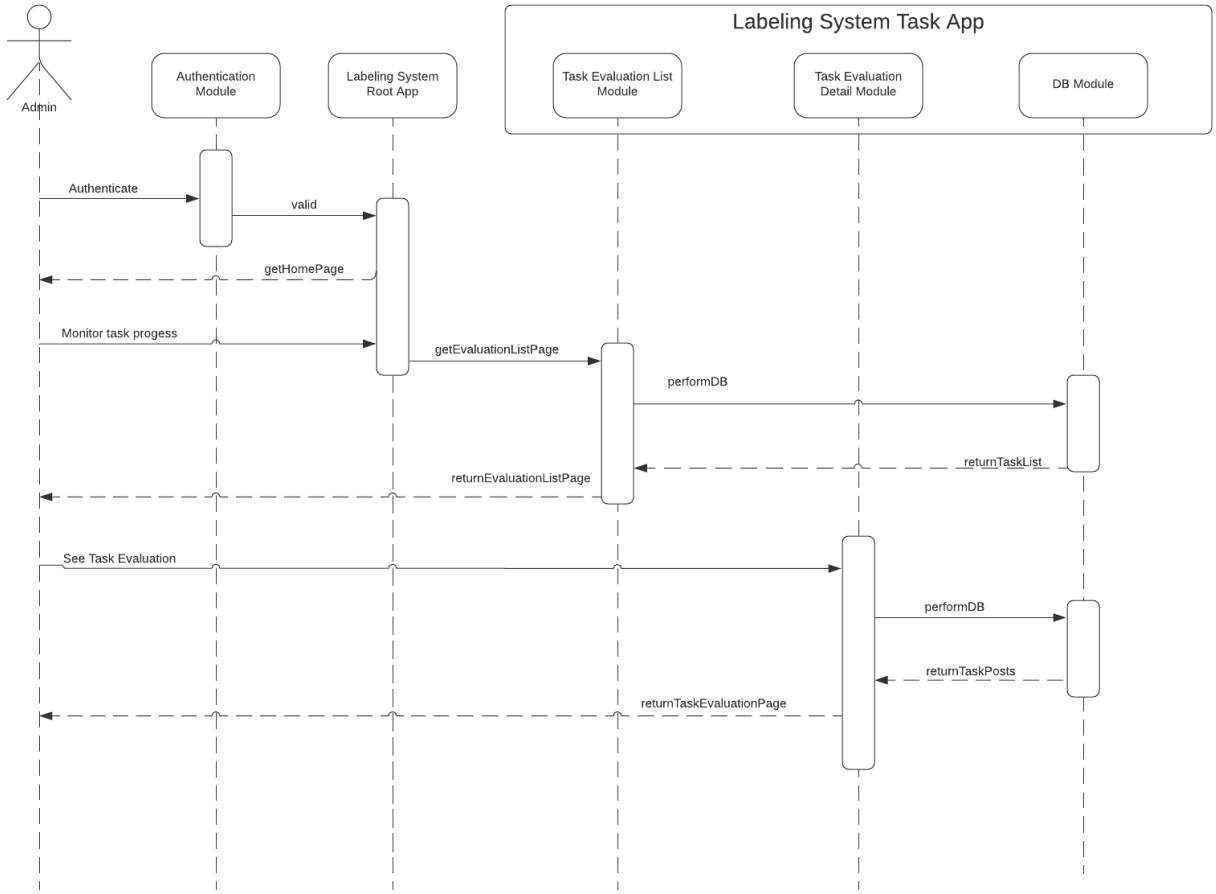


Fig 1.6: Sequence Diagram - Monitor Task and Evaluation

As shown in the above Fig 1.6

Task Evaluation List Module

- Performs DB query to get all tasks associated with the user along with the information about assigned labelers, responded labelers
- This information is parsed by the front end to display the information

Task Evaluation Detail Module

- Performs DB query to get all the posts associated with the task. These posts are then formatted and fed to inter rate reliability third party library "nlk" to compute Kappa Evaluation and Krippendorff's alpha and passed to the front end for display
- The statistical computations are written into CSV files which are used in frontend to facilitate download

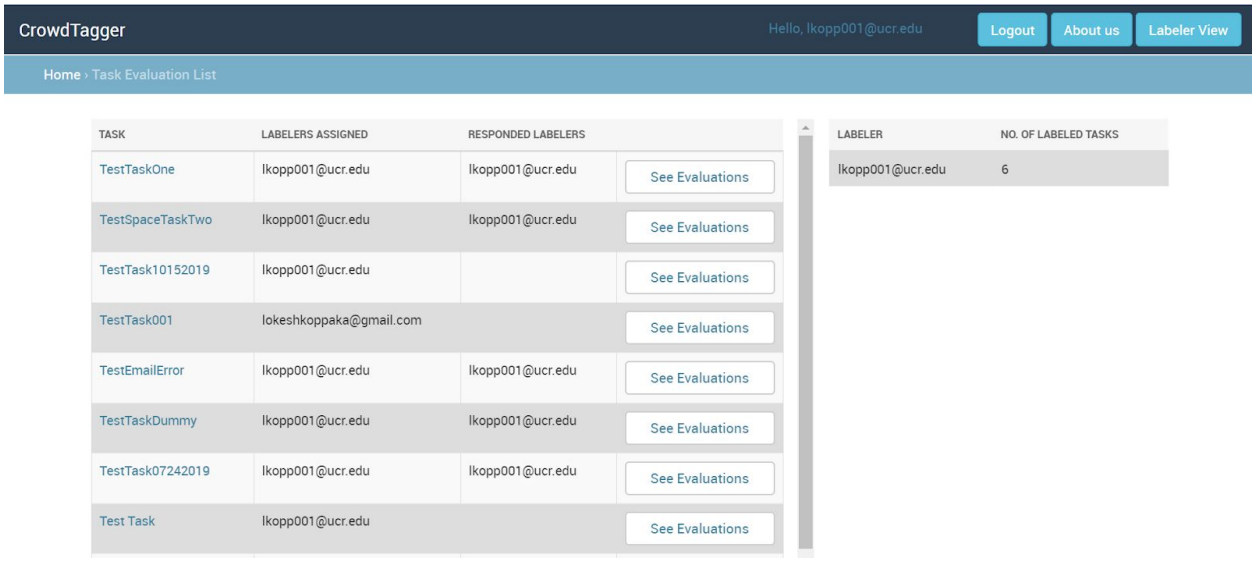
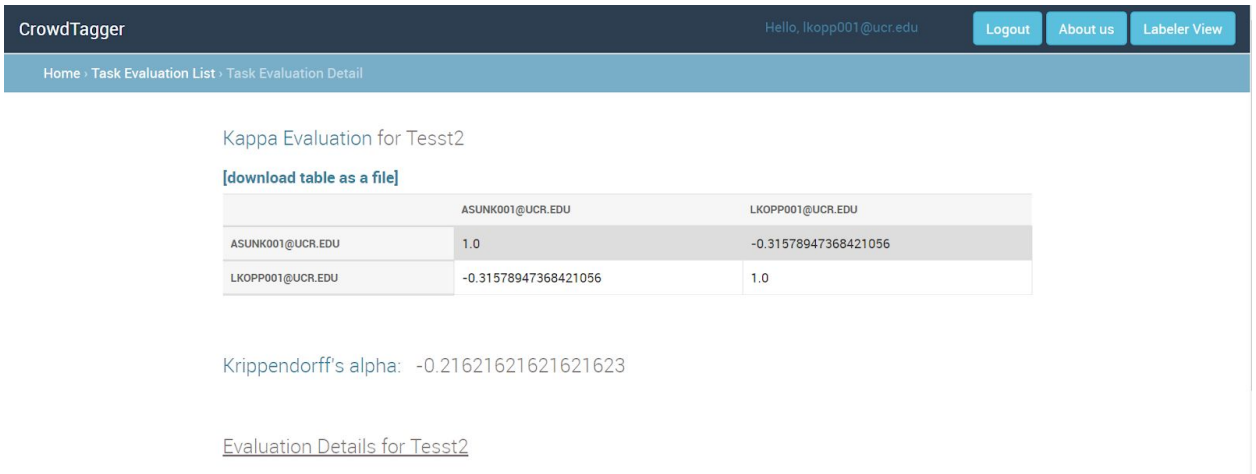


Fig 1.7: Task Evaluation List Page



[download table as a file]

POST	ASUNK001@UCR.EDU	LKOPP001@UCR.EDU	MAJORITY VOTE
Want to learn a little more about the origins of our company and how we stay successful? Check out this video: https://www.youtube.com/watch?v=VuukRMmh9-8	helpful	funny	helpful
My twitter has become so powerful that I can actually make my enemies tell the truth.	advertising	motivational	advertising
Had to bring out the shovel this morning and saw this online. Smart phone manufacturers get on it! Our backs will thank you. Drive safe everyone!	motivational	funny	motivational
Vary your tweets by either using text, link or image and with or without hashtag.	funny	helpful	funny
Hope you had a wonderful and safe holiday!	helpful	advertising	helpful

Fig 1.8: Task Evaluation Detail Page

Labeler Flows

Labeling Post

Labeling Post feature provides a way for the labeler to label the posts for a given task.

Labelling Steps:

- On the home page as shown in Fig 1.10 click “continue working” to continue labeling the task
- If the task has a prerequisite quiz to complete, the control is navigated to take quiz page, complete the quiz to start labeling
- Label the posts of the given task in the “take task page” as shown in Fig 1.11 till the progress is 100

Technical Details:

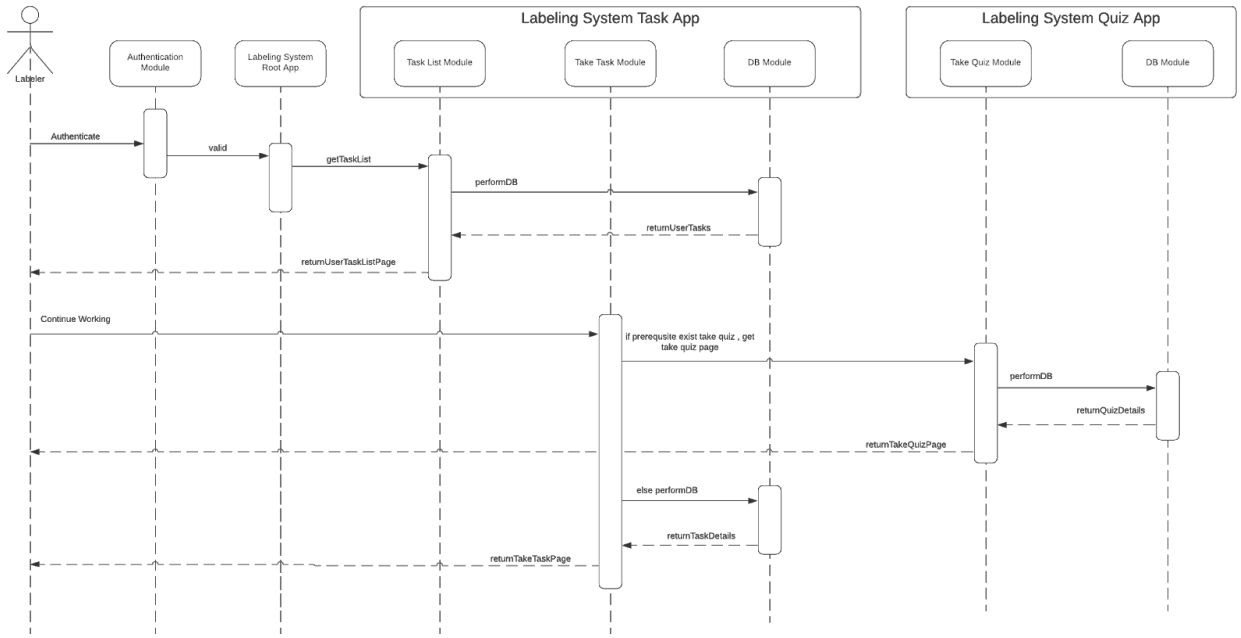


Fig 1.9: Sequence Diagram - Labeling Post

As shown in the above Fig 1.9

On home page load the task list module

- Performs DB query to get all the tasks assigned to the labeler by querying the participation table
- The queried records are then formatted such that it holds the task details along with the completion status
- The formatted data is used in the front end to display tasks as incompleted and completed tasks

On click on “Continue Working” the “Take Task” module

- Checks if there exists a prerequisite for the task.

- If yes, the control is moved to “Take Quiz” module which performs DB query to retrieve quiz posts and labels. These queried records are formatted and returned to the front end to display quiz post
- Else, the control is moved to “Take Task” module which performs DB query to retrieve task posts and labels. These queried records are formatted and returned to the front end to display task post

- On labeling of each task post, quiz post the responses are stored in postResponse and quizResponse tables respectively

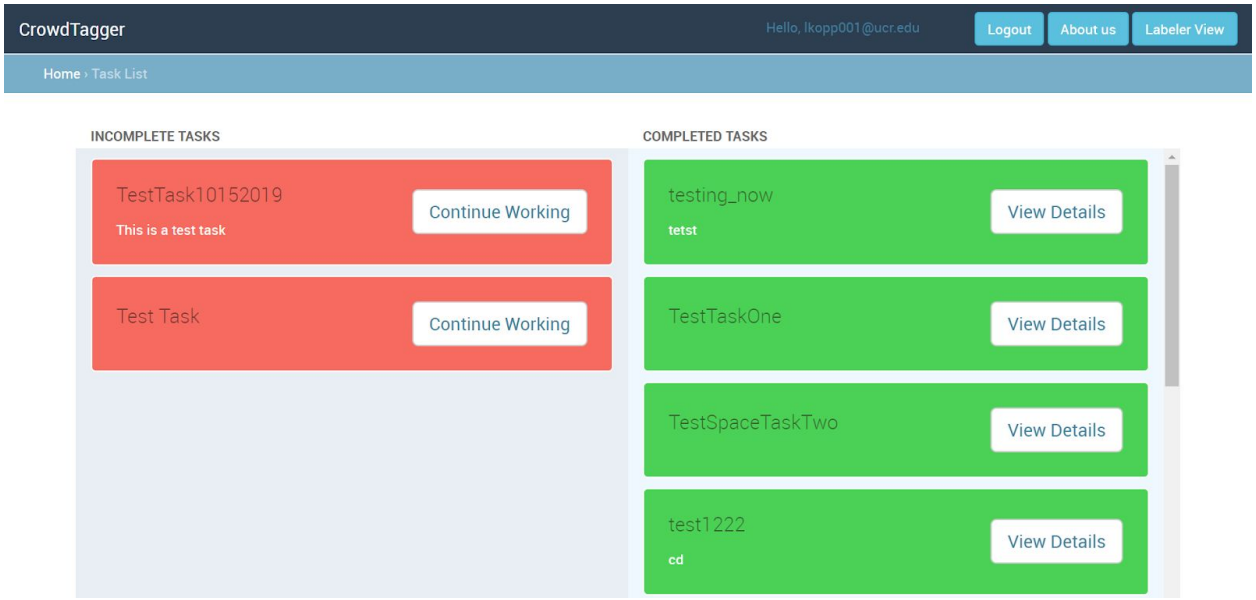


Fig 1.10: Labeler Home Page

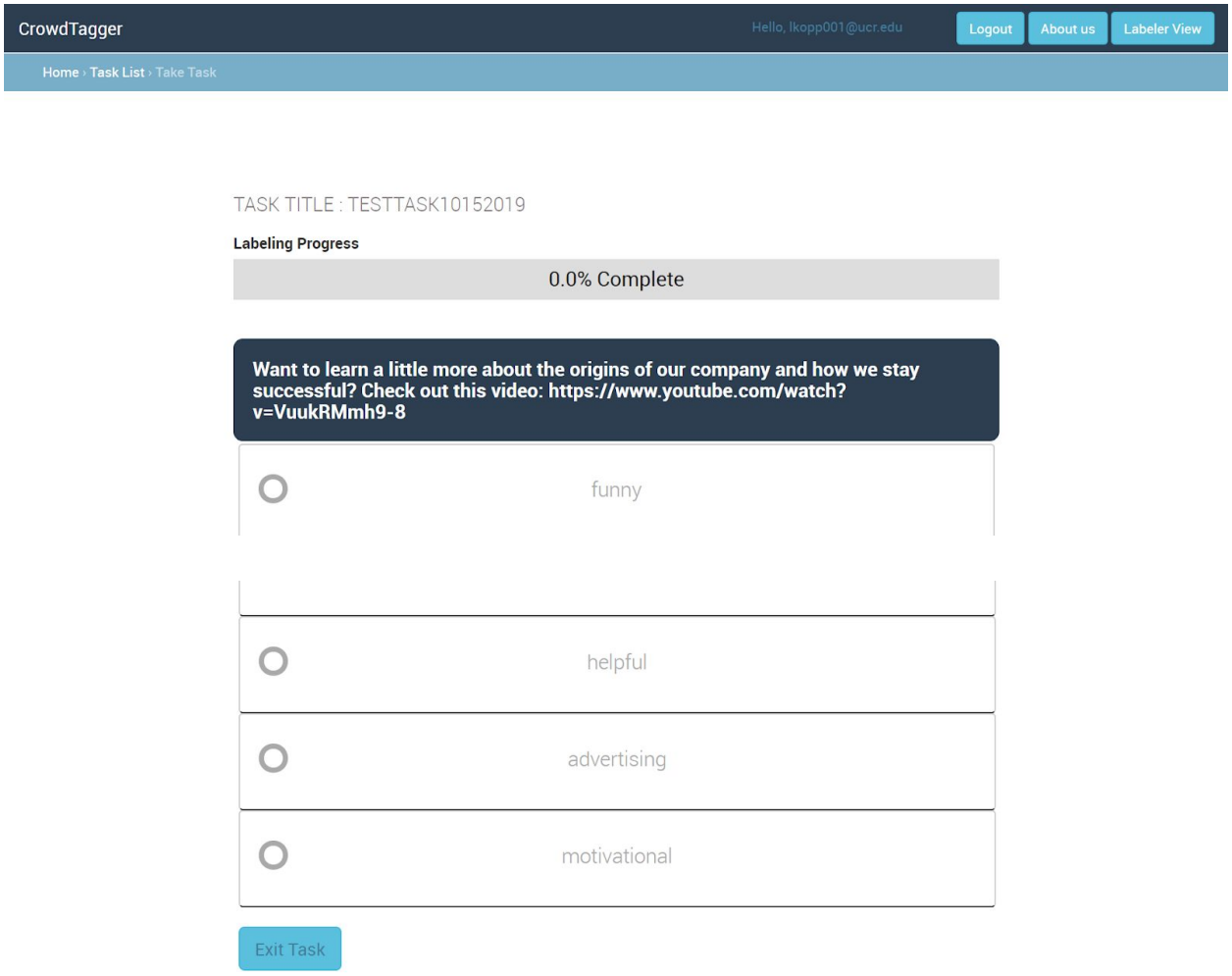


Fig 1.11: "Take Task" Page

Additional Information

[1] **Labeler:** a group of users who are assigned to a task, each labeler will be assigned some posts and need to respond with the most appropriate label

[2] **Post:** a list of posts that each labeler needs to respond to.

[3] **Label:** a list of possible labels that each labeler need to choose from

[4] **Quiz:** Taking a quiz is similar to taking a task, but in the quiz, each question has only one correct answer, labeler must answer the quiz and get enough credit in order to take the task

[5] **Kappa Evaluation:** is a statistic that is used to measure inter-rater reliability (and also Intra-rater reliability) for qualitative (categorical) items

[6] **Krippendorff's alpha:** Krippendorff's alpha is a reliability coefficient developed to measure the agreement among observers, coders, judges, raters, or measuring instruments drawing distinctions among typically unstructured phenomena or assign computable values to them. It emerged in the content analysis but is widely applicable wherever two or more methods of generating data are applied to the same set of objects, units of analysis, or items and the question is how much the resulting data can be trusted to represent something real

[7] **Django:** Django is a Python-based free and open-source web framework, which follows the model-template-view architectural pattern

[8] **Use case Diagram:** A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved

[9] **Sequence Diagram:** A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario

[10] **ER Model:** An entity-relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities