Bilkent University
Department of Computer Engineering
CS 491: Senior Design Project I
Fall 2021

Project Specifications Report

Project Name: *Laber*

Group Members:

Yiğit Gürses	21702746	yigit.gurses@ug.bilkent.edu.tr
Melisa Taşpınar	21803668	melisa.taspinar@ug.bilkent.edu.tr
Onur Oruç	21702381	onur.oruc@ug.bilkent.edu.tr
Emin Adem Buran	21703279	adem.buran@ug.bilkent.edu.tr
Mustafa Hakan Kara	21703317	m.kara@ug.bilkent.edu.tr

Supervisor: Assoc. Prof. Can Alkan
Innovation Expert: Ahmet Eren Başak
Jury Members: Assoc. Prof. Can Alkan
Asst. Prof. Dr. Shervin Arashloo
Asst. Prof. Dr. Hamdi Dibeklioğlu

Project Webpage: https://eminademburan.github.io/laber/

Table of Contents

1. Introduction 2
   - 1.1. Description 2
   - 1.2. Constraints 3
     - 1.2.1. Sustainability Constraints 3
     - 1.2.2. Implementation Constraints 3
     - 1.2.3. Economic Constraints 4
     - 1.2.4. Environmental Constraints 4
     - 1.2.5. Data Constraints 4
     - 1.2.6. Language Constraints 4
     - 1.2.7. Safety Constraints 5
     - 1.2.8. Ethical Constraints 5
   - 1.3. Professional and Ethical Issues 5

2. Requirements 6
   - 2.1. Functional Requirements 6
   - 2.2. Non-functional Requirements 9
     - 2.2.1. Usability 9
     - 2.2.2. Marketability 9
     - 2.2.3. Cost 9
     - 2.2.4. Reliability 9
     - 2.2.5. Scalability 9
     - 2.2.6. Efficiency 9
     - 2.2.7. Security 10
     - 2.2.8. Extendibility 10
     - 2.2.9. Portability 10

3. References 10

1. Introduction

With the coming of the information age and rise of social media, there is an increasing quantity of information available on the internet that can be taken advantage of [1]. Companies try to take advantage of this using natural language processing and human analysts but they both come with some disadvantages. Machine learning models try to approximate human expertise and always come with a margin of error [2], while processing a large amount of data with human experts can be very costly. In order to reduce this cost, crowdsourcing platforms such as Amazon's Mturk [3] can be used but such unspecialized platforms require expertise and investment on the side of the client. For these reasons, there exists a niche to be filled by specialized platforms that are easy to use for both the clients and human experts.

Laber will be a mobile based platform that aims to provide real time social media analysis by human experts. Clients will be able to crowdsource their work with minimal knowledge of the system and have access to a pool of human experts at all times. The obtained analytics will be available to the clients through our website. The experts will be able to do all of their work through a mobile application allowing them to work remotely and efficiently. Various features will be implemented to ensure the reliability of our experts and provide them the optimal environment to do their work. Gamification techniques will be utilized to incentivize regular work schedules.

In this report, we provide a brief description of the project and its constraints. Then, we touch on the possible professional and ethical issues related to implementation and future deployment. Finally, we give a detailed list of functional and nonfunctional requirements.

1.1. Description

Laber is a mobile based crowdsourcing platform that aims to provide clients with real time social media analysis through human experts. There will be two types of users, namely Clients and Experts. Clients refer to companies and institutions that need our services. Clients will interact with *Laber* through a browser. They can create tasks by specifying the following: a list of keywords and hashtags, a set of metrics and time interval. A metric, for example, can be a sentiment scale in the range [-10, 10]. *Laber* will automatically scrape posts that contain the given keywords or hashtags from websites, such as Facebook and Twitter, in real time. These posts will then be evaluated by human experts based on the given metrics. Experts will sign up through the mobile app with an SMS confirmation. They will be given social media posts to evaluate based on the metrics specified by the Client. *Laber* will then take these evaluations to plot graphs that show the change of metrics in time. Clients can then see these graphs through a browser and can choose to be notified when there is an instantaneous change in the metrics. The reliability of each Expert will be tracked through various means and their compensation will be correlated with this reliability score. All work will be done through the mobile app. The Experts will be compensated for their work by the Clients.

1.2. Constraints

1.2.1. Sustainability Constraints

- The platform will receive feedback from users regularly which will be taken into consideration for further updates.
- When more Experts and Clients start using the application, bug reports and analytics will be utilized to minimize bug encounters on the user end.

1.2.2. Implementation Constraints

- Twitter API [4] and Facebook API [5] will be used to obtain samples.
- The application will have both a web application for Clients and a mobile application for Experts.
- Github will be used as a version control tool to track the source code improvements.
- JavaScript, HTML, CSS, and PHP will be used to develop the web application.
- Java and SQL will be used for the back end.
- React Native [6] and Java will be used for the mobile application.

- Amazon Web Services [7] will be used for data storage and real time processing.

1.2.3. Economic Constraints

- The money that Clients will pay and the Experts will earn will be determined by the pricing of similar applications in the market. The pricing will be semi-dynamic to ensure the availability of human experts at all hours of the day.
- Releasing the application on the Google Play Store will have a one time cost [8].
- Memory and processing costs on the back end will incur a recurring fee which will increase as the number of users grows.

1.2.4. Environmental Constraints

- The mobile application and back end implementation will be as efficient as possible to reduce memory and energy usage.
- Experts will do all of their work on the mobile application allowing them to work remotely, eliminating the carbon footprint of work commutes.

1.2.5. Data Constraints

- Clients will provide keywords and hashtags to be searched for.
- Posts to be evaluated at the request of Clients will be scraped from Twitter and Facebook in full context.
- Instead of scraping data from the website within a time interval, Clients can also choose to upload their own data to be evaluated.

1.2.6. Language Constraints

- The application will be available in English. Thus, it can be used by users who know English from all over the world.
- The application will be implemented in a way that makes it easy to add new languages.

1.2.7. Safety Constraints

- In order to be able to use our mobile application, human experts will have to accept that they will not use our application while driving or doing other dangerous activities.

1.2.8. Ethical Constraints

- We will be handling large amounts of sensitive data for both Clients and Experts so it is critical that we have a secure back end that ensures there will be no leakage.
- Anyone with access to a mobile device and internet will be able to use the app as a human expert.

1.3. Professional and Ethical Issues

Laber will collect and store the information required to process transactions (e.g., name, surname, contact information, bank account). There should not be any bugs or security flaws in the platform that may cause a leakage of sensitive data.

If any personal information is to be shared with a third party, the users will be informed and their permission will be required. This way, the privacy of users will be assured. Furthermore, Clients will not be allowed to see the analytics that were requested and paid for by other clients.

If a software product that is to be used in the project requires a licence, either a licence will be obtained or an open source alternative will be used in its place. If any portion of the code is taken from another source, the original writers will be referenced.

2. Requirements

2.1. Functional Requirements

- Users can sign up as a Client or Expert. Clients can register through a browser. Experts can register through the mobile application with an SMS confirmation.
    - Experts can modify their user name and profile picture
- Experts can invite their friends with an invite link to receive rewards
- Experts can send friend requests to other Experts using their phone number or username.
- Experts can accept pending friend requests.
- Experts can see their friends list and their friends' statistics.
- The Clients will be able to create task streams:
    - Can specify keywords and hashtags to be searched for.
    - Can specify a set of metrics and their intervals/categories. For example: {sentiment: [-1, 1], sarcasm: (serious, sarcastic, mixed)}.
    - Can specify which websites should be searched for from a predetermined list.
    - Can specify a time interval. The beginning and the end of the interval could be any two date-times. The end of the interval can also be indefinite, which means new posts will keep getting scraped until the task stream is stopped manually.
    - Instead of scraping data from the website within a time interval, Clients can also choose to upload their own data to be evaluated, provided that the data is in the same format as ours.
    - Can specify the type of Experts allowed to work on the task stream. (region, language, age etc.)
- The system will scrape samples from social media platforms according to the task stream specifications.
    - Each sample will consist of a single social media post.
    - Full context of the post will be scraped so that it can be provided to the Expert. The context includes:
        - Profile picture
        - User name

        - Text
        - Links
        - Any image/video/audio included
        - Engagement statistics (likes, reblogs etc.)
        - Comments
    - New samples will be placed in the task queue to be evaluated by Experts.
- The system will distribute the samples across the currently active Experts.
    - The same sample will be given to multiple Experts. Ideally, to a mix of highly reliable Experts and newly registered ones so that the new ones can be evaluated based on the comparison of their results.
- In order to ensure the availability of Experts at all times:
    - The pay rate will be correlated to the ratio of number of samples in the task queue to number of active Experts.
    - Experts will be notified when the pay rate reaches a threshold that the experts can set for their own accounts.
- The Experts will evaluate the samples based on the given metrics.
    - The sample will be shown in full context and all work will be done through the mobile app.
    - Experts will gain points for each task completed based on their reliability score. Experts with higher reliability scores will earn more points.
    - Certain numbers of tasks will be considered milestones. (10, 100, 1000 etc.)
        - Milestone progress of all Experts will be reset each day.
        - Experts that reach a milestone will earn bonus points.
- The system will validate expert evaluations with the following methods:
    - The same sample will be distributed to multiple Experts and their results will be cross-checked.
    - If the results happen to have a high variance, the Experts who generated them will engage in a discussion through the system’s built-in voice chat feature, to reach a consensus.
    - As an alternative to voice chat, there will be discussion threads.
    - Machine learning will be utilized as follows:

        - A pretrained NLP model will be deployed via transfer learning to evaluate each sample.
        - The model will keep learning as new human expert evaluations are obtained.
        - Once the accuracy reaches a certain threshold, the model's evaluations will be used to check human experts' reliability.
- The system will keep track of the Experts' reliability scores.
    - Occasionally, Experts will be given mock samples with known labels to check their reliability.
    - For samples without known labels, their answers will be compared to other Experts' answers.
    - Based on the points above, their reliability score will change depending on the quality of their work.
- Obtained Expert evaluations will be used to generate graphs that show the change of metrics in time. The Clients will be able to view these graphs and other analytics through a browser.
    - Evaluations from Experts with high reliability scores will have higher weights in the final results.
    - Samples that had high engagement in the site they were taken from will have a bigger impact on the final results.
    - Frequently occurring words and their correlations to the metrics will be provided.
- Clients can specify critical metrics. If an instantaneous change in a critical metric is detected, the client will be notified with an automatic mail.
- Experts can withdraw their points as real life currency or exchange them for various benefits like discounts on certain shops.
- Experts will be placed in weekly leagues. Experts that just signed up will be placed in the lowest league. Experts in a league will be ranked based on the amount of points they earned that week. The top 20% in a league will be promoted to the next league, and bottom 30% will be demoted to the previous league at the end of each week. Experts will earn bonus points at the start of each week based on their current league.

2.2. Non-functional Requirements

2.2.1. Usability

- Clients should be able to create tasks and see the analytics through a browser without any technical knowledge.
- Experts should be able to do all of their work through our mobile application without any technical knowledge.
- The application should be efficient and run smoothly to not interrupt the workflow of experts.

2.2.2. Marketability

- Human experts are incentivized to use our app through monetary compensation and gamification.
- *Laber* provides the Clients real time access to social media analysis through human experts, with minimal cost.

2.2.3. Cost

- All services will be free of charge except the payments of human experts, which will be provided by the clients.

2.2.4. Reliability

- There should not be any failures that cause loss of work or payments.
- The application should run smoothly to not interrupt Experts' workflow.

2.2.5. Scalability

- The back end should be implemented keeping in mind the fact that the data to be handled can become arbitrarily large.

2.2.6. Efficiency

- The application should be efficient enough to run on low end mobile devices.

- The back end should be efficient in terms of memory and processing complexity in order to minimize the costs associated with servers.

2.2.7. Security

- SMS confirmation will be required for human experts when signing up, in an effort to eliminate bots and malicious actions.
- There should not be any bugs or security flows that may cause the leakage of sensitive information.

2.2.8. Extendibility

- The implementation should allow the addition of new task types. These might include image/audio transcription, text completion and other services that might be compatible with our crowdsourcing infrastructure.

2.2.9. Portability

- The system should support various mobile devices and web browsers.
- The mobile application should be designed to be easily ported to different operating systems.

3. References

[1] Lohr, Steve. "The Age of Big Data." *The New York Times,* 11 Feb. 2012, https://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html. Accessed 5 Oct. 2021.

[2] Cavaioni, Michelle. "Machine Learning: Causes of error." *Medium*, 1 Feb. 2017, https://medium.com/machine-learning-bites/machine-learning-causes-of-error-87ff372cd5be. Accessed 5 Oct. 2021.

[3] "Amazon Mechanical Turk." *Amazon Mechanical Turk*, https://www.mturk.com/. Accessed 5 Oct. 2021.

[4] "Docs." *Twitter Developer Platform,* https://developer.twitter.com/en/docs. Accessed 9 Oct. 2021.

[5] "Facebook For Developers." *Facebook For Developers*, https://developers.facebook.com/. Accessed 9 Oct. 2021.

[6] "React Native." *React Native,* https://reactnative.dev/. Accessed 7 Oct. 2021.

[7] "Amazon Web Services." *aws*, https://aws.amazon.com/tr/. Accessed 7 Oct. 2021.

[8] "How to Upload an App to Google Play Store?" *Appintentiv,* https://appinventiv.com/blog/how-to-submit-app-to-google-play-store/#:~:text=There%20is%20a%20one%2Dtime,your%20name%2C%20country%20and%20more. Accessed 9 Oct. 2021.