Bilkent University

Department of Computer Engineering

CS 491: Senior Design Project I

Fall 2021

Analysis 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

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 first provide a brief description of the system we are proposing. Then, we elaborate specifically on its functional, non-functional and pseudo requirements, as well as the system models. In the system models section, we provide the scenarios corresponding to using our program, our use-case model, object & class model, dynamic models, and finally a description of our user-interface using mock-up's. Towards the end of the report, before providing a glossary and our references, we talk about other analysis elements such as the consideration of various factors in engineering design, risks & alternatives, project plan, ensuring proper teamwork, ethics and professional responsibilities, planning for new knowledge and learning strategies.

2 Proposed System

2.1 Overview

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 called labels, and a time interval. A label, 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 labels. Experts will sign up through the mobile app with an SMS confirmation. They will be given social media posts to evaluate based on the labels specified by the Client. *Laber* will then take these evaluations to plot graphs that show the change of labels 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 labels. 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.

2.2 Functional Requirements

- Users can sign up as a Client or an Expert. Clients can register through a web 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.

2.3 Nonfunctional Requirements

2.3.1 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.
- Experts should be able to communicate with each other well using the discussion threads and voice chats. To maintain the quality of the audio communication as well as the usability of the voice chat, there should not be any large delays or loss of data.
- The application should be efficient and run smoothly to not interrupt the workflow of experts.

2.3.2 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.

2.3.3 Cost

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

2.3.4 Reliability

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

2.3.5 Scalability

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

2.3.6 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.

2.3.7 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.

2.3.8 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.

2.3.9 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.

2.4 Pseudo Requirements

- Target Platform: Our mobile application will preferably be available both on IOS & Android.
- Version Control: We will be using Git as our version controller.
- Implementation Language: For the web application section of our project, we will be using PHP as our language of implementation. For the mobile part, we will be using Javascript.
- Framework: We will be using React Native [4] for our mobile application.

2.5 System Models

2.5.1 Scenarios

2.5.1.1 Sign Up

Scenario Name: applicationEnrollment
Participating actor instances: Ali: Expert
Flow of events:

1. Ali tries to register to the application by typing name, surname, e-mail address, phone number, age, region, language, password and friend's invitation link.
2. Ali registers with his email address and phone number. *Laber* sends an SMS and email confirmation codes to Ali.
3. Ali enters the submission codes correctly and is registered successfully.

2.5.1.2 Create Task

Scenario Name: taskCreation
Participating actor instances: Ayşe: Client
Flow of events:

1. Ayşe comes to create a task page. She enters details such as keywords, hashtags, and metrics about the task to be created.
2. She chooses Facebook, and Twitter to be the websites that the posts will be searched for.
3. She determines the intervals during which the task will stay active.
4. Ayşe chooses the experts who will evaluate tasks later to be 20-25 years old, binary gender, native english speaker, and from Europe.
5. As all required fields have been filled by Ayşe, the task is created by *Laber* when Ayşe submits the task.

2.5.1.3 Voice Chat

Scenario Name: voiceChatForPost
Participating actor instance: Iggy: Expert
Flow of events:

1. While Iggy is waiting for a task to be assigned to him on the main page, he notices that a task which requires joining a voice channel is assigned to him indicating that the results he contributed earlier have a high variance.
2. To enter the voice chat, Iggy gives Laber permission to access his microphone. He then waits for other Experts to enter the voice chat.
3. When every participant is ready, Iggy starts discussing, by his microphone, the previous results with other Experts who contributed to the previous results too. They continue to discuss until the timer expires (3 mins).
4. Considering the discussions, Iggy makes a new decision and re-submits his evaluation to the system.

2.5.1.4 Discussion Thread

Scenario Name: discussionThreadForPost

Participating actor instance: Billie, Chilly, Emily: Expert
Flow of events:

1. While Billie is waiting for a task to be assigned to him on the main page, she notices that a task which requires joining a discussion thread is assigned to her indicating that the results he contributed earlier have a high variance.
2. Billie accepts to join the discussion thread and starts reading the previous comments on the topic by Chilly and Emily.
3. After reading other comments, she contributes to the discussion and posts a comment. Billie, Chilly, and Emily keep on discussing the previous results.
4. Based on the discussions, Billie makes a new evaluation and submits it to the system.

2.5.1.5 Post Assessment

Scenario Name: postAssestment
Participating actor instance: Iggy: Expert
Flow of events:

1. While Iggy is waiting for a task to be assigned to him on the main page, he notices that a task is assigned to him.
2. Iggy starts to evaluate the post according to the metrics given to him sequentially.
3. Iggy wants to change his mind, and goes back to the previous metrics and changes his choices.
4. After evaluating his own decisions, he thinks that he is ready to submit his choices. He submits his evaluation and starts to evaluate the new post.

2.5.1.6 Withdraw Funds

Scenario Name: fundWithdrawal
Participating actor instance: Dawes: Expert
Flow of events:

1. Dawes goes to his profile page and decides to withdraw some funds. He selects the withdrawal option and is directed to the withdrawal page.
2. Out of 2 options, Crypto and Bank Transfer, he selects to withdraw his funds to a crypto address.

3. The fields that are required to be filled for the completion of the withdrawal action are displayed to Dawes by *Laber*. He then fills the fields (address, network, and amount to be withdrawn). While filling the fields, he pays attention to the warnings about minimum transaction amount that he can withdraw and transaction fee that is required for the system to finalize the transaction.
4. Dawes confirms the transaction and is notified by *Laber* that the transaction has been completed successfully.

2.5.1.7 Go Profile

Scenario Name: profileVisit
Participating actor instance: Dawes: Expert
Flow of events:

1. Dawes visits his profile and sees information about himself such as his balance, score, league, profile photo.
2. He decides to change his profile picture and select a new picture from his gallery.
3. He wants to add a friend to his friend's list, he chooses the add friend option from his profile page and is directed to the Add Friend page. He sends a person a friendship request and goes back to his profile page.
4. Then he wants to see his friends' statistics. He selects the option to list his friend's statistics and after viewing the statistics, he goes back to the profile page.

2.5.2 Use Case Model

In this subsection we provide the descriptions corresponding to the functionalities of our program, as well as our use-case diagram.

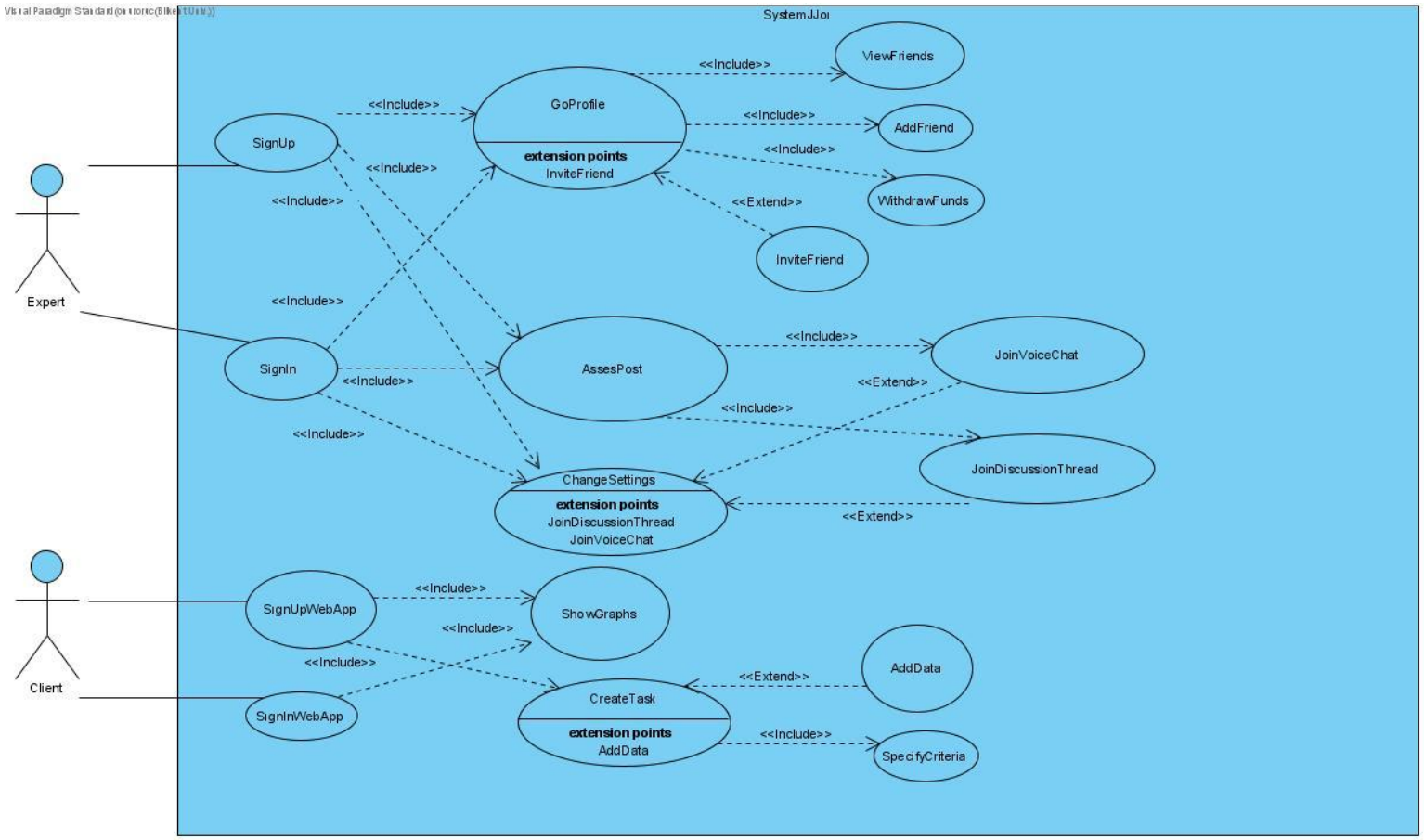


Figure 1: Use Case Diagram

2.5.2.1 Sign Up

Use case name: Sign Up

Participating actors: Expert

Flow of events:

1. The Expert opens *Laber*.

2. The Expert fills the fields (name, surname, phone, email address, age, region, language, password, and invitation link, if available) in the form required to sign up.

3. The Expert submits the form. If the form meets requirements, a verification email is sent to Expert's email account.

4. The Expert is asked to enter the verification code in a pop-up dialog. Otherwise, Expert is asked to fill the form again.

5. The Expert account is created after the verification code is accepted.

Entry conditions:

- The Expert opens *Laber*.
- The Expert is not already signed in.

Exit conditions:

- The Expert closes *Laber*.

Quality requirements:

- The response to the submitted information should be available to Expert within 5 seconds.

2.5.2.2 Sign In

Use case name: Sign In

Participating actors: Expert

Flow of events:

1. The Expert opens *Laber.*

2. The Expert fills the fields (email and password) and submits the form.

3. If the information provided is correct, Expert is logged in. Otherwise, Expert is asked to fill the form again.

Entry conditions:

- The Expert opens *Laber.*
- The Expert is not already signed in.

Exit conditions:

- The Expert closes *Laber.*

Quality requirements:

- *Laber* should direct the Expert to the main page within 3 seconds after the form is approved.

2.5.2.3 Assess Post

Use case name: Assess Post

Participating actors: Expert

Flow of events:

1. Post to be evaluated is shown to the Expert.

2. If the post was evaluated before and has a high variance, *Laber* invokes the JoinVoiceChat or JoinDiscussionThread use case based on which one is appropriate for the post.

3. If the post is new, the Expert is expected to evaluate the post based on the given metrics.

4. The Expert finishes evaluation for all the given metrics and submits the evaluation.

5. *Laber* evokes The Asses Post use case for the Expert for the evaluation of a different post.

Entry conditions:

- The Expert is logged in *Laber*.
- The Expert is assigned at least one post.

Exit conditions:

- The Expert closes *Laber*.
- The Expert is logged off.
- The Expert finishes all the posts that need to be evaluated.

Quality requirements:

- After the Expert finishes evaluation of a post, a new post, if any, should be visible to the Expert within 1 seconds.

2.5.2.4 Join Voice Chat

Use case name: JoinVoiceChat

Participating actors: Expert

Flow of events:

1. The Expert joins the Voice Chat by giving *Laber* access to the microphone.

2. Using a microphone, The Expert discusses the results, containing high variance, of a previously evaluated post by the Expert and other Experts.

3. The Expert re-evaluates the post and submits the evaluation.

4. *Laber* evokes The Asses Post use case for the Expert for the evaluation of a different post.

Entry conditions:

- The Expert allows *Laber* to access the microphone.
- The Expert is on the main page and assigned to join a voice chat.

Exit conditions:

- The Expert closes *Laber.*
- All the Experts in the voice chat submit a new evaluation for the post.

Quality requirements:

- The Experts that are supposed to join the voice chat should join within 10 seconds. Otherwise, *Laber* appoints different Experts to the post in replacement for the Experts that did not join the voice chat.

2.5.2.5 Join Discussion Thread

Use case name: JoinDiscussionThread

Participating actors: Expert

Flow of events:

1. The Expert joins the Discussion Thread.

2. The Expert reads the comments posted on the thread.

3. The Expert writes a comment to the thread and posts it.

4. The Expert re-evaluates the task and submits the evaluation.

5. *Laber* evokes The Asses Post use case for the Expert for the evaluation of a different post.

Entry conditions:

- The Expert is on the main page and assigned to join a discussion thread.

Exit conditions:

- The Expert submits a new evaluation for the post.

Quality requirements:

- The Expert should be able to post a comment to the thread within 1 second after the submission of the comment.

2.5.2.6 Go Profile

Use case name: Go Profile

Participating actors: Expert

Flow of events:

1. The Expert chooses to go profile.

2. *Laber* displays the profile of the Expert.

3. The Expert looks at her/his score, friends, friend requests, league, balance in profile page.

Entry conditions:

- The Expert should be on a page in which the navigation bar at the bottom is active.

Exit conditions:

- The Expert chooses to assess posts.
- The Expert chooses to go settings.
- The Expert chooses to go announcement.

Quality requirements:

- *Laber* should list the profile picture, current balance, score and league of the Expert.
- *Laber* should display buttons which will enable Experts to show their friends, friend requests and withdraw money.

2.5.2.7 Withdraw Funds

Use case name: WithdrawFunds

Participating actors: Expert

Flow of events:

1. The Expert chooses the Withdraw option from the profile.

2. *Laber* directs the user to the Withdrawal page.

3. The Expert chooses the withdrawal method.

4. If the Expert wants to withdraw to the bank account, the Expert fields the required fields (account name, IBAN, withdrawal amount). If the Expert wants to withdraw to a crypto address, the Expert enters address, network, and amount.

5. If the information submitted by the Expert meets the requirements of the system, withdrawal is approved. If the information is erroneous or does not meet the system requirements, Expert fills the required fields again.

6. *Laber* responds by presenting a dialog for the Expert to enter email verification code.

7. The Expert submits the verification code.

8. If the verification code is correct, *Laber* executes the transaction, notifies the Expert and directs the Expert to the Profile Page.

9. If the verification code is incorrect, *Laber* prompts the Expert to enter correct code until the correct code is submitted.

Entry conditions:

- The Expert is logged into *Laber.*
- The Expert chooses the Withdrawal option on the Profile Page.

Exit conditions:

- *Laber* approves the withdrawal and finishes the transaction.

Quality requirements:

- *Laber* should send the verification code within 5 seconds.
- *Laber* should execute the transaction operation within 2 seconds after the correct verification code is submitted.

2.5.2.8 Add Friend

Use case name: AddFriend

Participating actors: Expert

Flow of events:

1. The Expert activates the Add Friend option from the Profile Page.

2. The Expert searches for an Expert by name.

3. The Expert sends a friend request to the selected Expert.

4. *Laber* sends the request to the corresponding Expert.

Entry conditions:

- The Expert chooses the Add Friend option from the Profile Page.
- The Expert is logged into *Laber*.

Exit conditions:

- The Expert sends a friend request to an Expert.

Quality requirements:

- *Laber* should list the Experts that have the name searched by the Expert within a second.

2.5.2.9 View Friends

Use case name: ViewFriends

Participating actors: Expert

Flow of events:

1. The Expert activates the Add Friend option from the Profile Page.

2. *Laber* displays the friends of the Expert along with their names, leagues and scores.

Entry conditions:

- The Expert chooses the Show Friends option from the Profile Page.
- The Expert is logged into *Laber.*

Exit conditions:

- The Expert chooses the option to go back to the Profile Page.

Quality requirements:

- *Laber* should list the information for friends within a second.

2.5.2.10 View Tutorial

Use case name: View Tutorial

Participating actors: Expert

Flow of events: **1.** The Expert is logged into *Laber* for the first time.

2. *Laber* displays the parts of the application one by one. It informes the Expert about what is the utility of leagues, how to evaluate tasks, what to do in a voice chat or a discussion thread, etc.

3. The Expert confirms that the tutorial is understood.

Entry conditions:

- The Expert is logged into *Laber* for the first time.

Exit conditions:

- The Expert chooses the option to go to another page.
- The Expert sends an acknowledgement that the tutorial is understood.

Quality requirements:

- The Expert should understand what they will do in *Laber* clearly.

2.5.2.11 Invite Friends

Use case name: InviteFriends

Participating actors: Expert

Flow of events:

1. The Expert activates the Add Friend option from the Profile Page.

2. *Laber* displays the invitation number.

3. The Expert share his/her invitation number with his/her friends.

Entry conditions:

- The Expert chooses the Add Friends option from the Profile Page.
- The Expert is logged into *Laber*.

Exit conditions:

- The Expert chooses the option to go back to the Profile Page.

Quality requirements:

- *Laber* should list the invitation number within a second.

2.5.2.12 Change Settings

Use case name: ChangeSettings

Participating actors: Expert

Flow of events:

1. The Expert chooses to change his/her settings.

2. *Laber* displays current settings.

3. The Expert can adjust his/her settings according to his/her needs.

Entry conditions:

- The Expert chooses the settings option from the Navigation Bar below of the page.
- The Expert is logged into *Laber*.

Exit conditions:

- The Expert chooses another option from the Navigation Bar.

Quality requirements:

- The expert should access his/her setting while assessing a post.

2.5.2.13 Sign Up Web App

Use case name: Sign Up Web App

Participating actors: Client

Flow of events:

1. The client opens *Laber* in his/her web browser.

2. The client fills the fields (name, surname, phone, email address, age, region, language, password, and invitation link, if available) in the form required to sign up.

3. The client submits the form. If the form meets requirements, a verification email is sent to Client's email account.

4. The Client is asked to enter the verification code in a pop-up dialog. Otherwise, Client is asked to fill the form again.

5. The Client account is created after the verification code is accepted.

Entry conditions:

- The Client opens *Laber* in his/her web browser.
- The Client is not already signed in.

Exit conditions:

- The Client closes *Laber*.

Quality requirements:

- The response to the submitted information should be available to the Client within 5 seconds.

2.5.2.14 Sign In Web App

Use case name: Sign In Web Page

Participating actors: Client

Flow of events:

1. The Client opens *Laber*.

2. The Client fills the fields (email and password) and submits the form.

3. If the information provided is correct, Client is logged in. Otherwise, Client is asked to fill the form again.

Entry conditions:

- The Client opens *Laber*.
- The Client is not already signed in.

Exit conditions:

- The Client closes *Laber*.

Quality requirements:

- *Laber* should direct the Client to the main page within 3 seconds after the form is approved.

2.5.2.14 Show Graphs

Use case name: Show Graphs

Participating actors: Client

Flow of events:

1. The Client goes to the graphs page about the statistics of pre-created tasks .

2. The Client receives information about his tasks in detail.

Entry conditions:

- The Client chooses to go to the graphs page.
- The Client is signed in.

Exit conditions:

- The Client chooses to go create a task page, account page or payment page.

Quality requirements:

- *Laber* should direct the Client to the graphs page within 3 seconds.

2.5.2.15 Create Tasks

Use case name: Create Tasks

Participating actors: Client

Flow of events:

1. The Client goes to the create task page to create a task.

2. If The Client wants to use an existing dataset for the task, the Add Data use case is evoked by *Laber*. Otherwise. The Client specifies the task details, websites to be used, start and end date, and expert preferences.

3. The Client fills the fields (email and password) and submits the form.

Entry conditions:

- The Client chooses to create a task.
- The Client is signed in.

Exit conditions:

- The Client chooses to go to the graph page, account page or payment page.

Quality requirements:

- *Laber* should direct the Client to create a task page within 3 seconds.

2.5.2.16 Add Data

Use case name: Add Data

Participating actors: Client

Flow of events:

1. The Client chooses to create a task.

2. The Client chooses to add data about the task such as the urls of posts taken from social media and submit the form.

Entry conditions:

- The Client should create a task.
- The Client is signed in.

Exit conditions:

- The Client closes *Laber*.

Quality requirements:

- *Laber* should add data taken from the Client to its database within at most 1 minute.

2.5.2.17 Specify Criteria

Use case name: Specify Criteria

Participating actors: Client

Flow of events:

1. The Client chooses to create a task.

2. The Client fills the fields (task details, website which *Laber* will take post, time interval, experts preferences) and submits the form.

Entry conditions:

- The Client should create a task.
- The Client is signed in.

Exit conditions:

- The Client closes *Laber*.

Quality requirements:

- *Laber* should add the criteria taken from the Client to its database within at most 5 seconds.

2.5.3 Object and Class Models

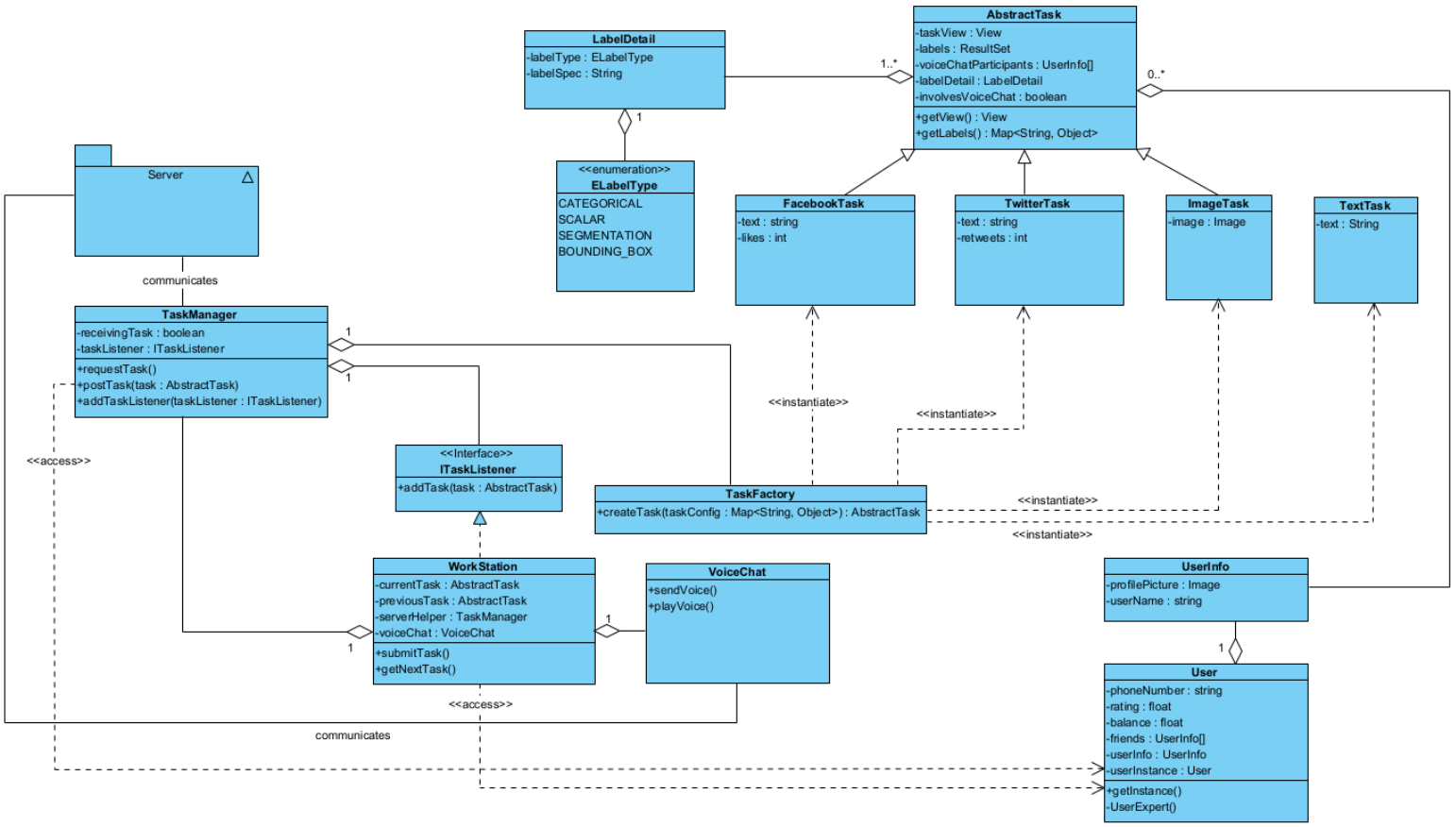


Figure 2: Class Diagram For the Mobile Application

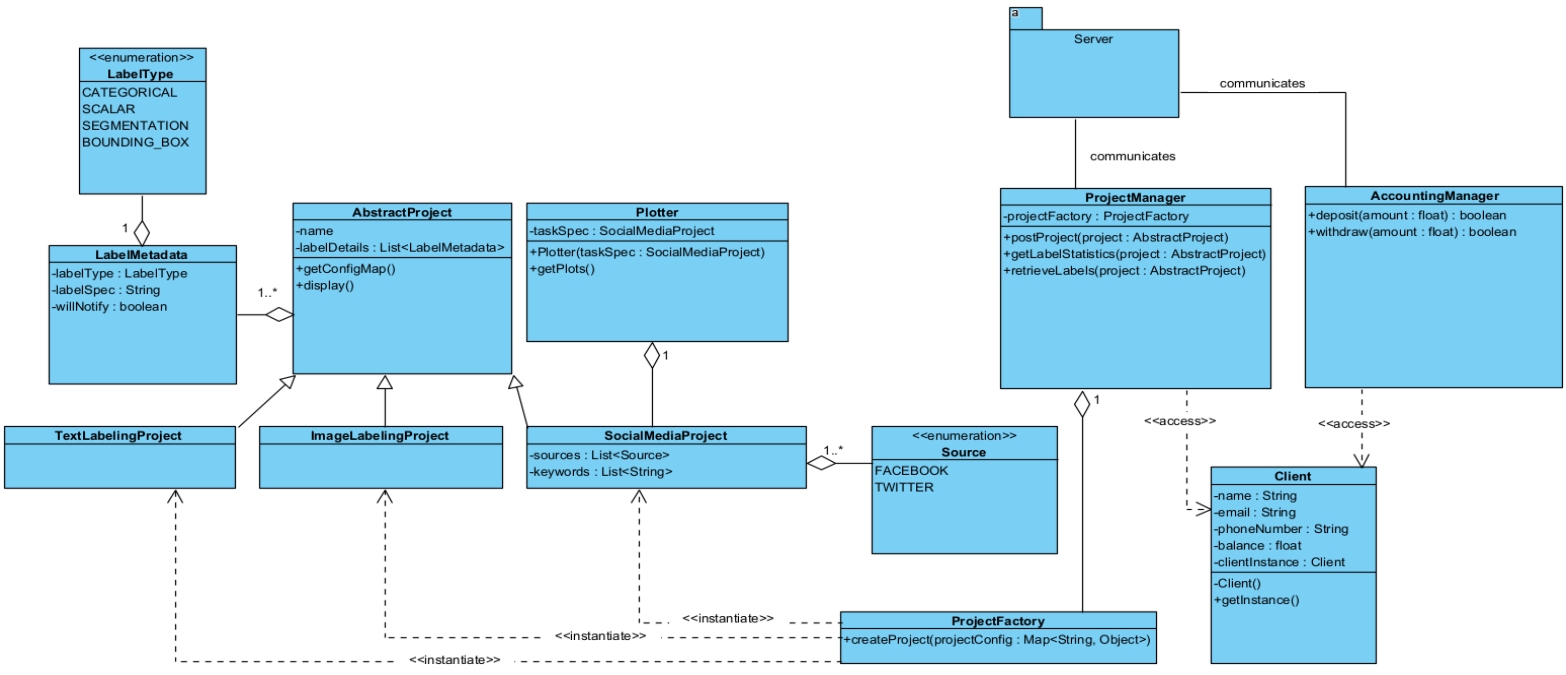


Figure 3: Class Diagram For the Website

2.5.4 Dynamic Models

2.5.4.1 State Diagrams

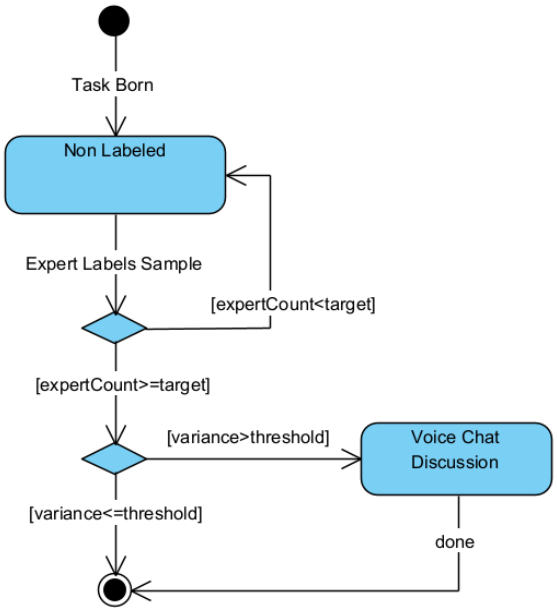


Figure 4: State Diagram of a Task

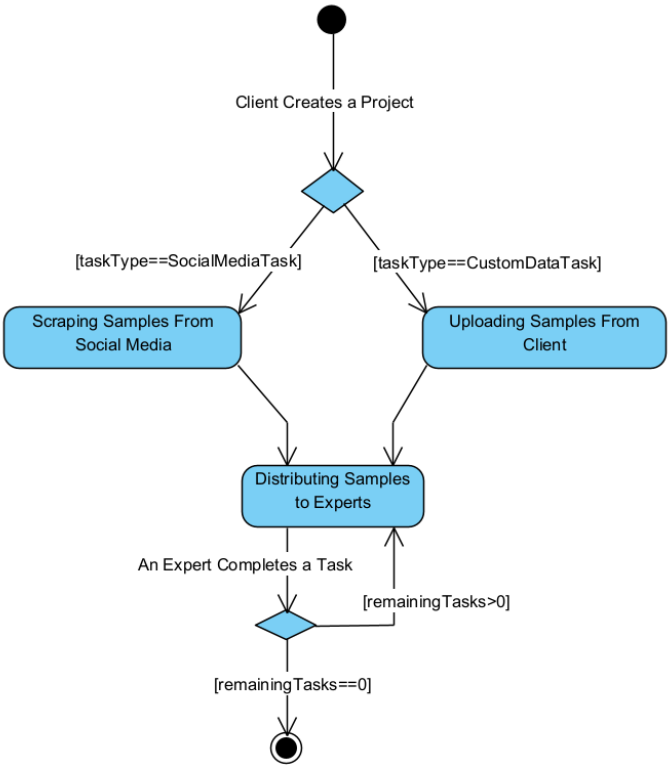


Figure 5: State Diagram of a Project

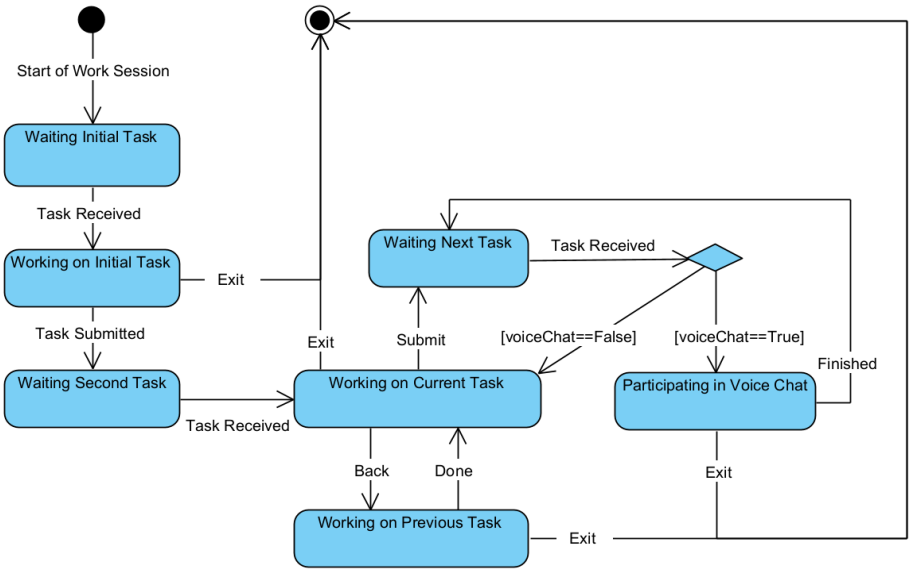


Figure 6: State Diagram of WorkStation

2.5.4.2 Activity Diagrams

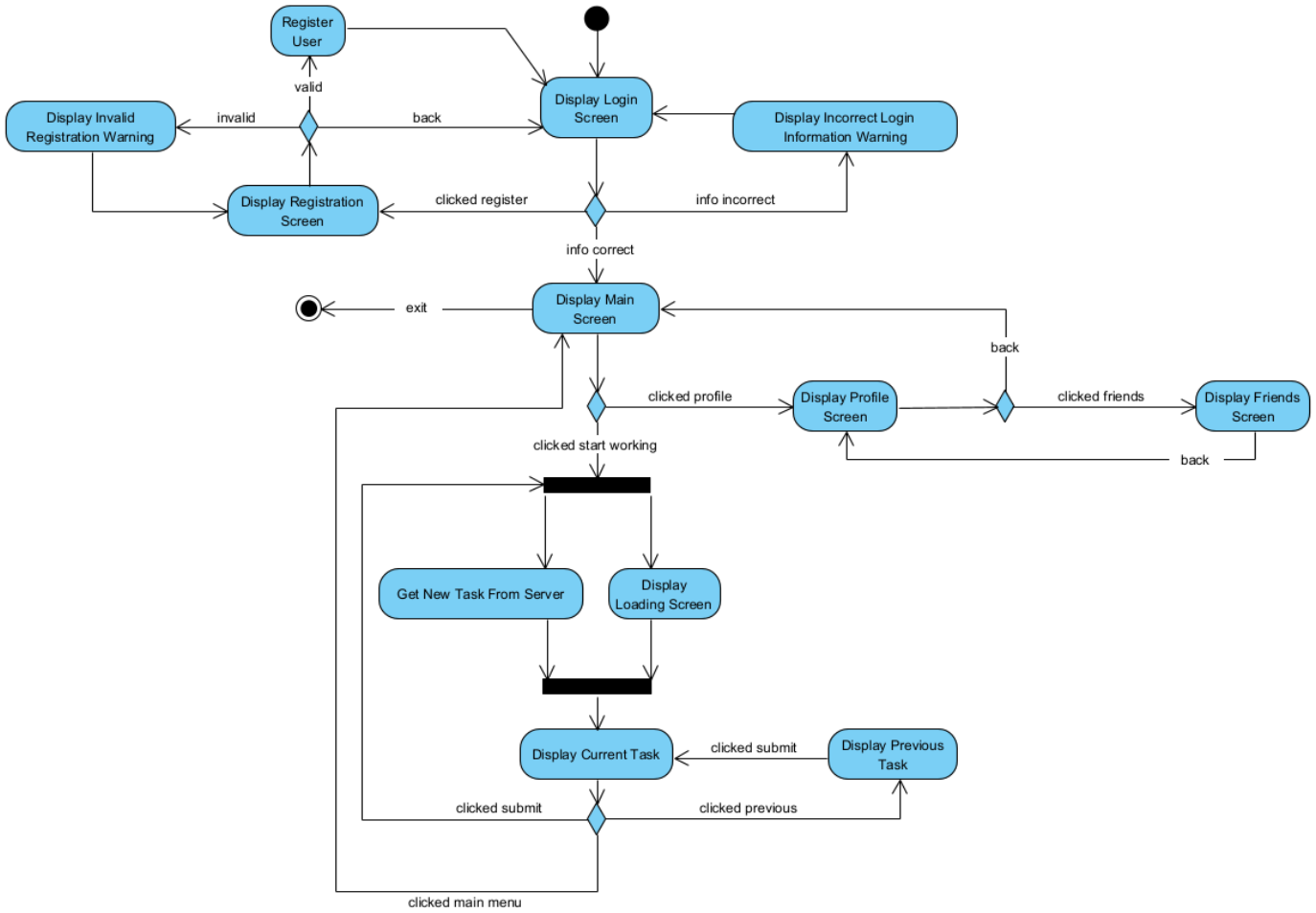


Figure 7: Activity Diagram of Mobile Application

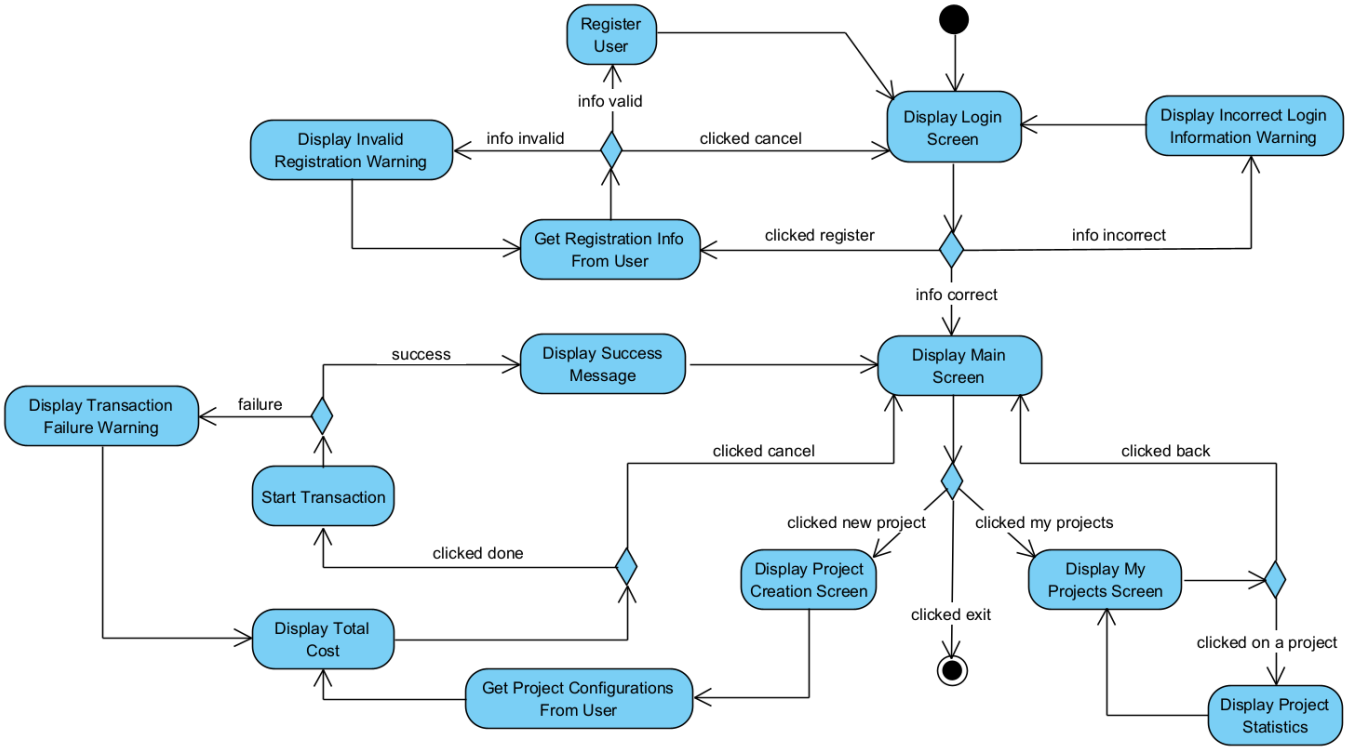


Figure 8: Activity Diagram of the Website

2.5.4.3 Sequence Diagrams

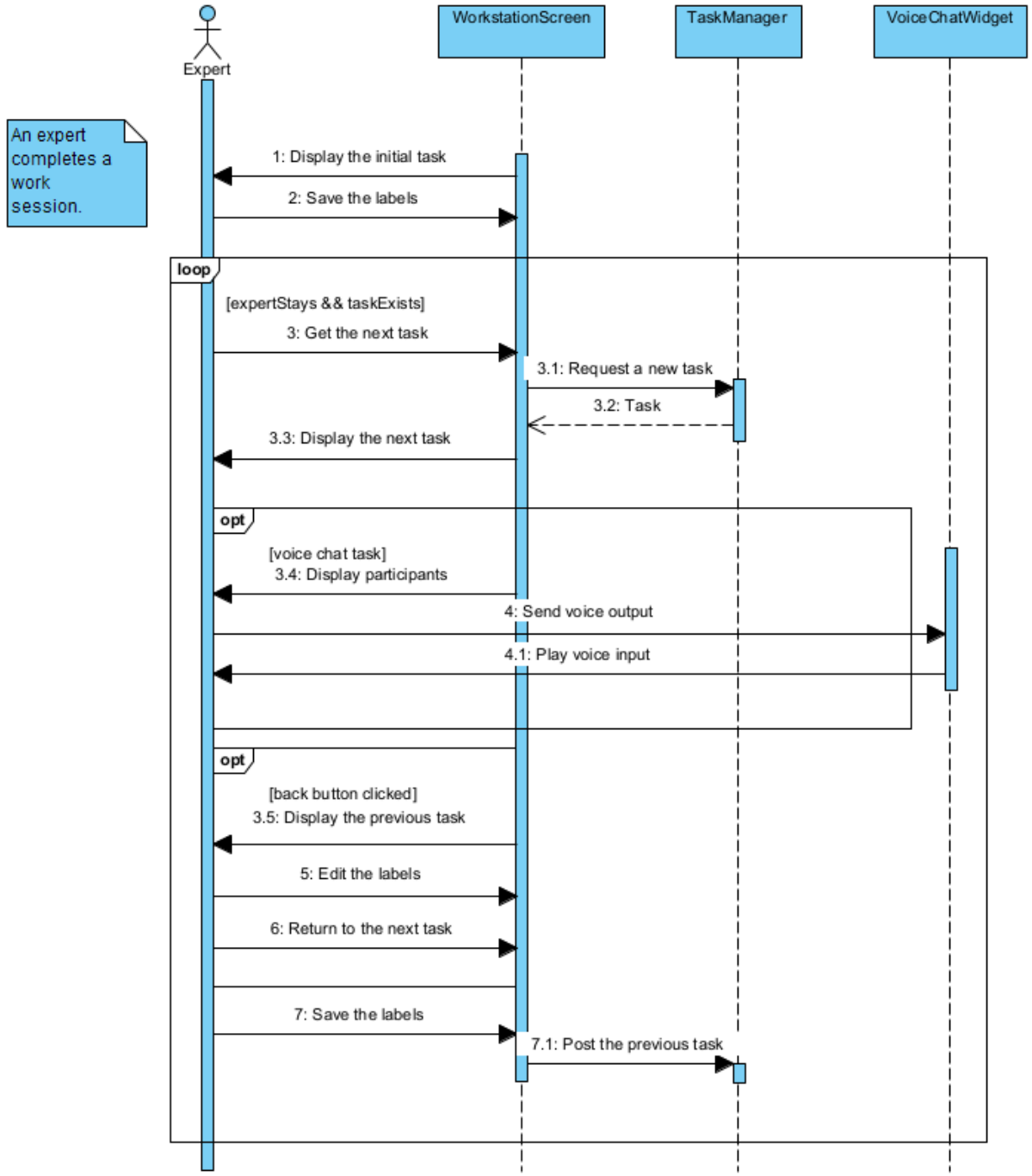


Figure 9: Sequence Diagram for a Work Session of an Expert

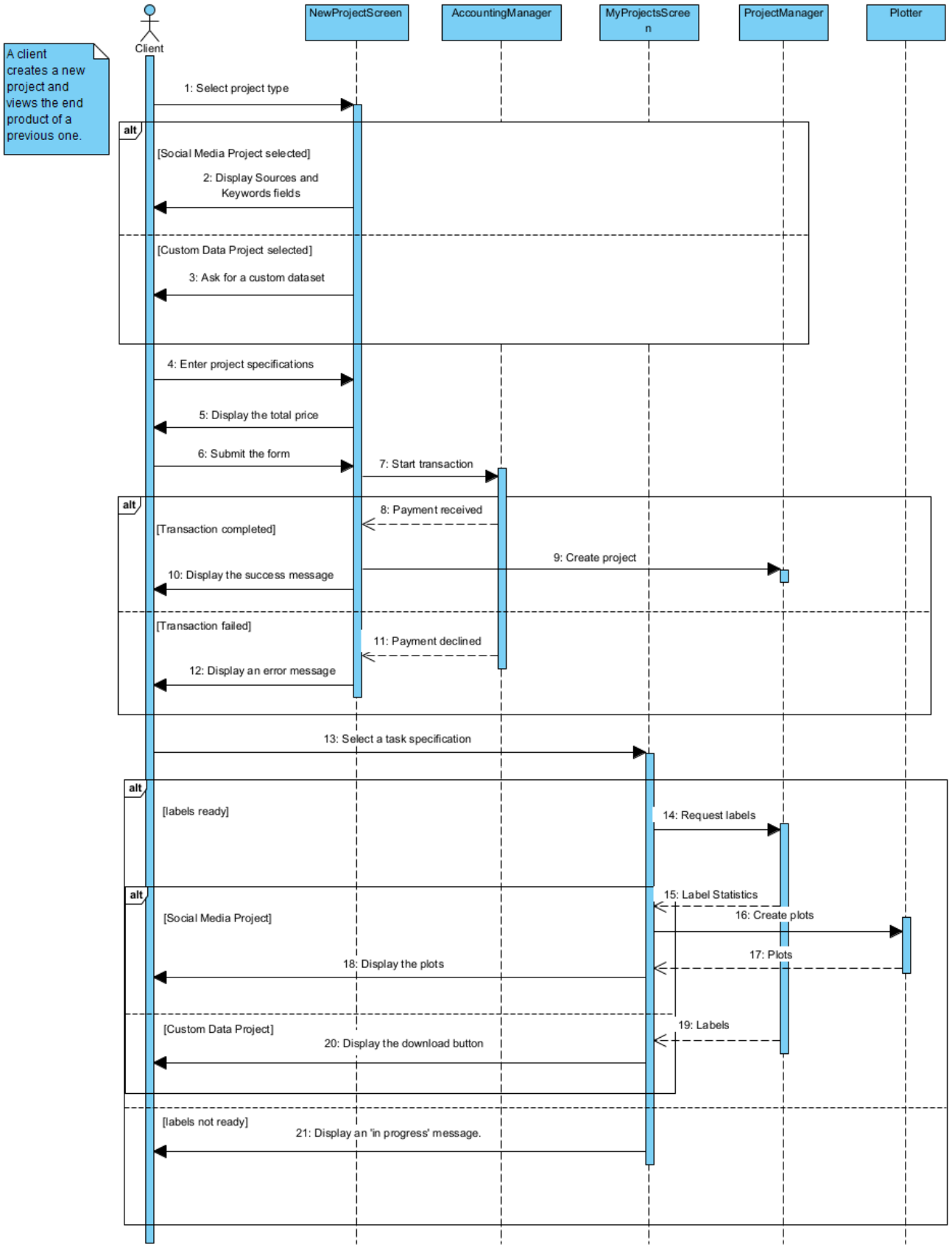
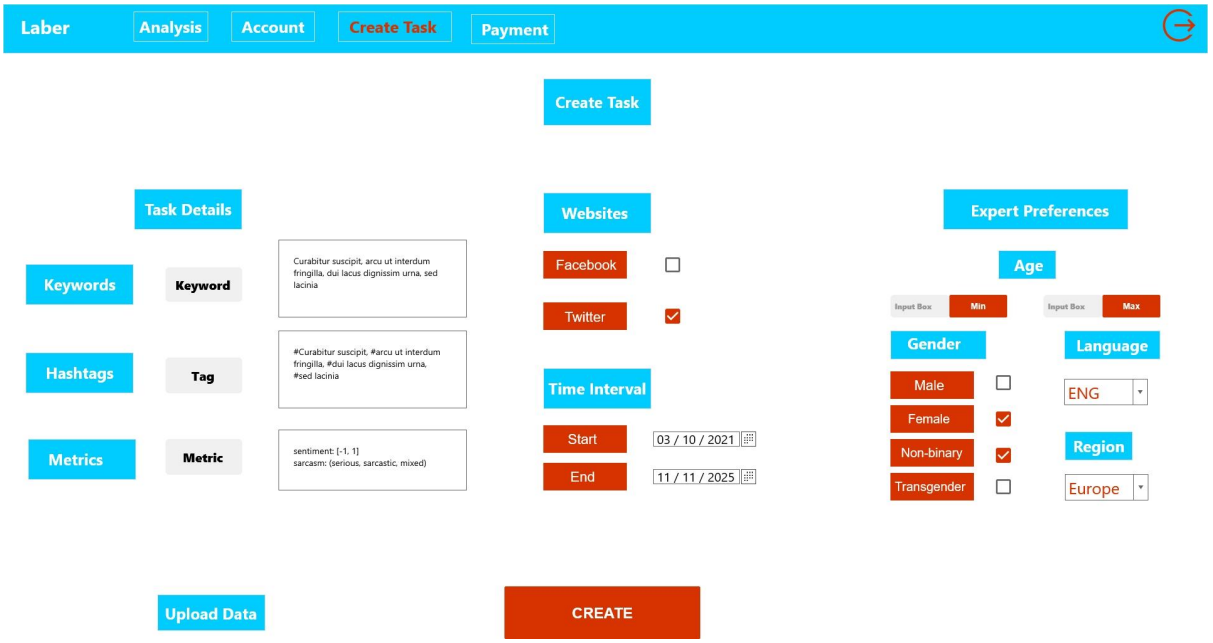


Figure 10: Sequence Diagram of Client Creating a Project and Then Viewing an Existing Project

2.5.5 User Interface - Navigational Paths and Screen Mock-ups

2.5.5.1 Create Task Mock-up (for Website)

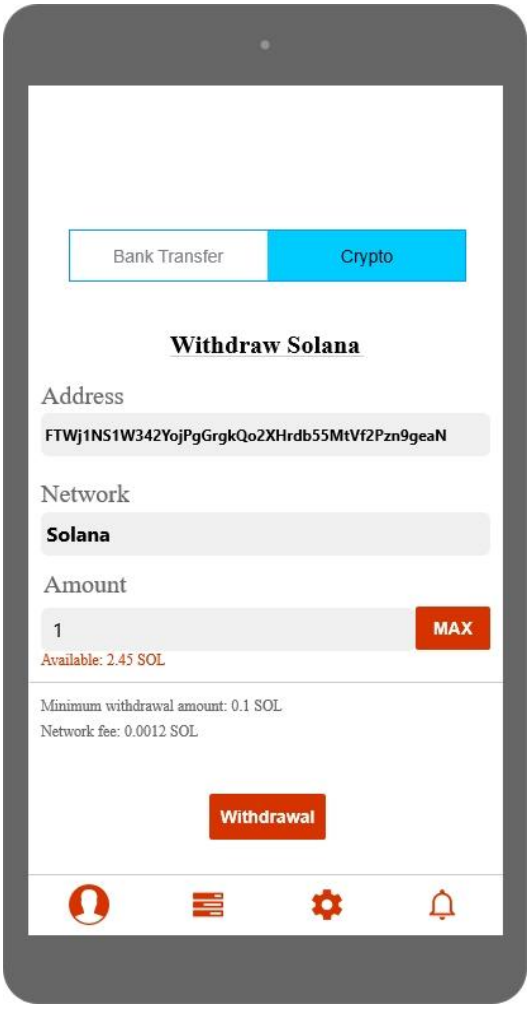


In this page, the Client creates a task by specifying its details. S/he enters keywords, hashtags, metrics, start and end dates, and expert preferences such as age, gender, language, region.
The Client either chooses Facebook or Twitter for obtaining data based on the metrics or uploads an existing dataset to the system.

2.5.5.2 Graphs of Tasks Mock-up (for Website)

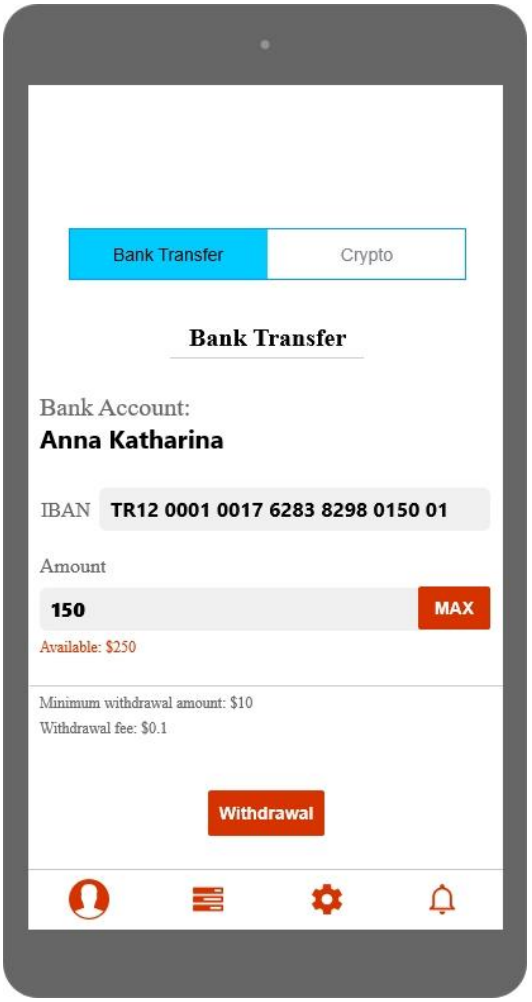
In this page, the Client views the statistical graphs about some metrics related to a previous task that was created by the Client and evaluated by the Experts.

2.5.5.3 Withdrawal Fund Page - Crypto



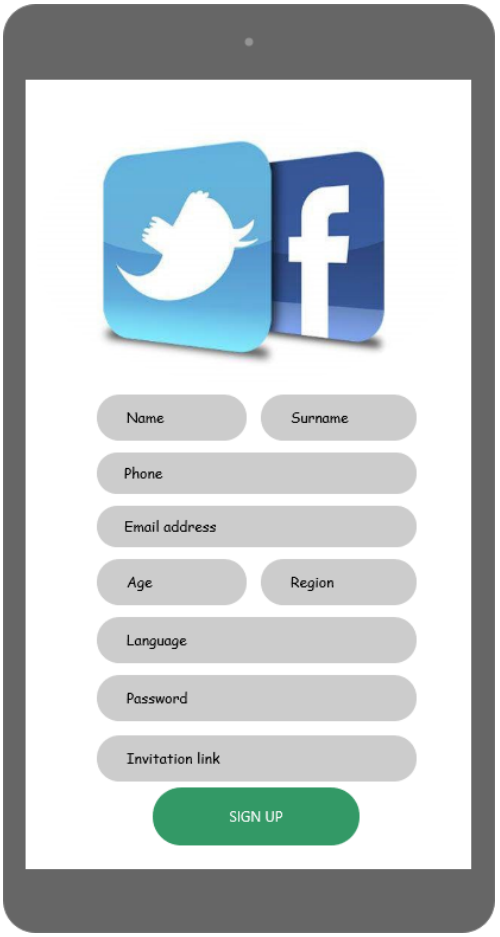
In this page, the Expert can withdraw funds to a crypto wallet address by filling the address, network and amount fields.

2.5.5.4 Withdrawal Fund Page - Bank Transfer



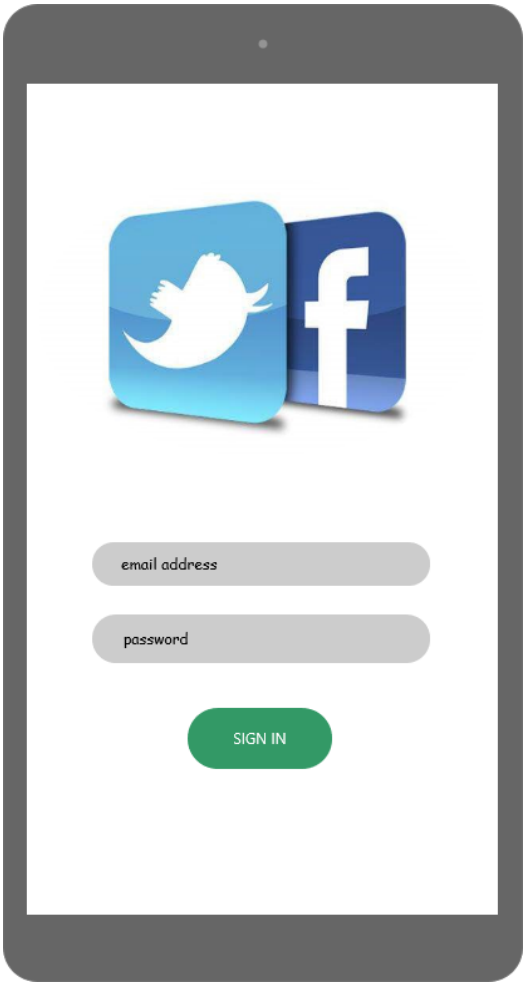
In this page, the Expert can withdraw funds to a bank account by filling account and amount fields.

2.5.5.5 Sıgn Up Mobile App



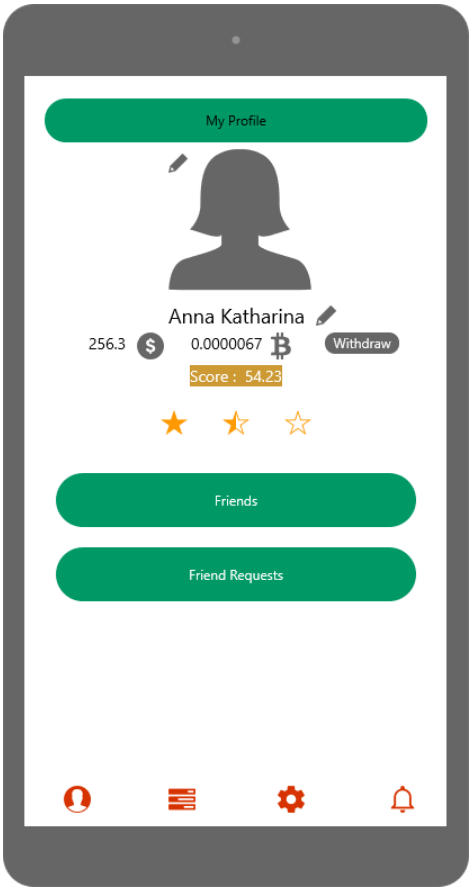
People can sign up to *Laber* by entering name, surname, phone, email address, age, region, language, password and invitation link. All fields should be filled except the invitation link to sign up.

2.5.5.6 Sıgn In Mobile App Mock-Up



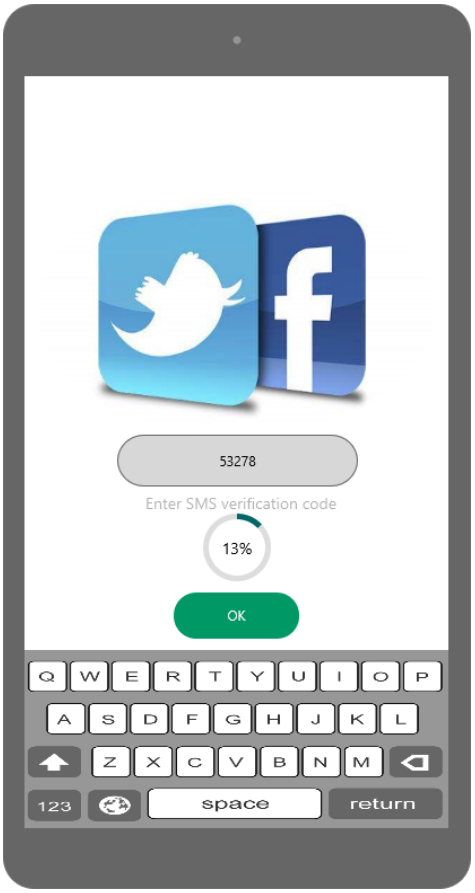
The Client can sign into *Laber* by entering his/her email address and password correctly.

2.5.5.7 Expert Profile Mock-up



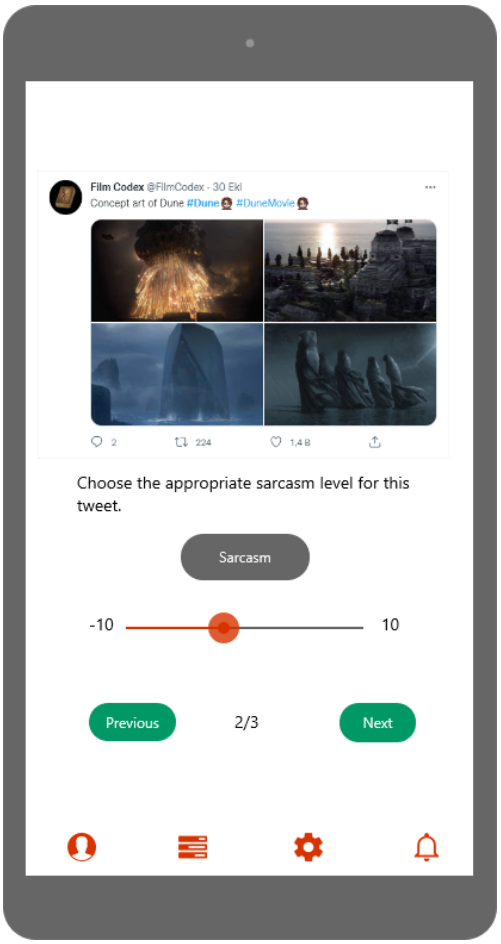
An Expert can see his/her avatar, balance, league, and score on the profile page. S/he can view his/her friend's statistics by pressing the Friends button. S/he can also view friend requests coming from other Experts by pressing the Friend Requests button.

2.5.5.8 SMS Verification Mock-up



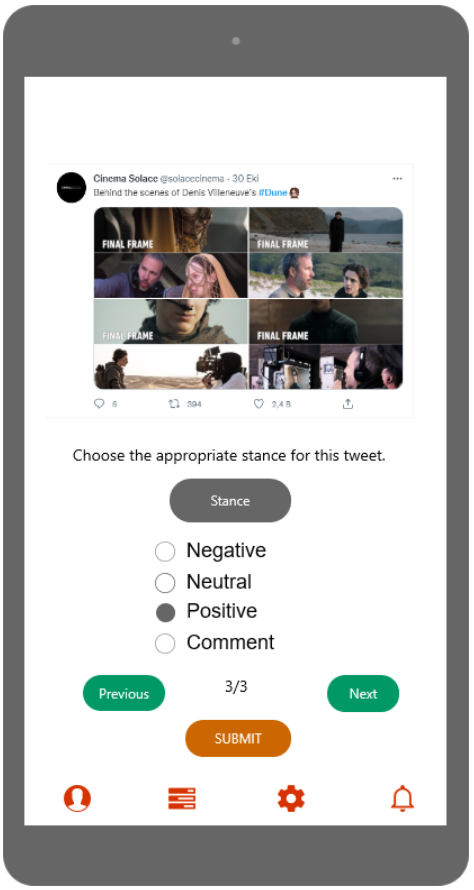
SMS verification code is sent to Expert while both signing up and signing into *Laber*. The Client is supposed to enter the correct verification code within the time limit. Otherwise, the verification code expires.

2.5.5.9 Post Assessment - Scalar



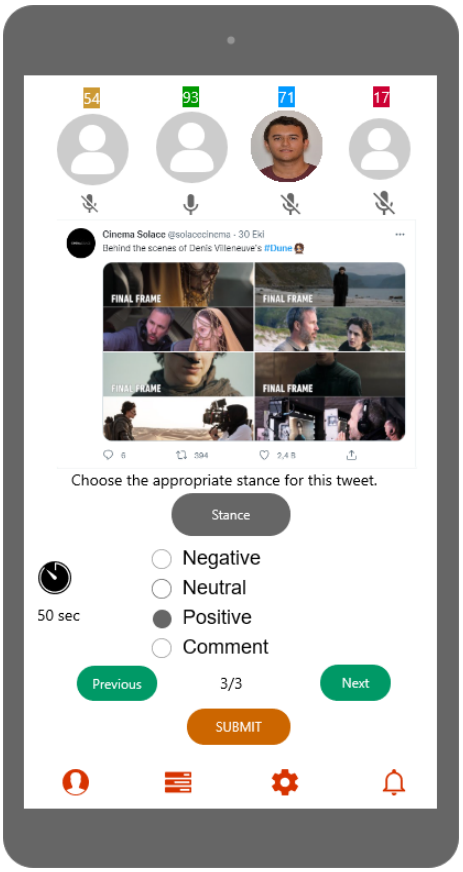
In this page, the Expert sees the post. Posts may contain combinations of text, image, and voice. The Expert, based on the post and given metrics, makes an evaluation. After finishing all the metrics, the Client submits the evaluation to the system.

2.5.5.10 Post Assessment - Non-scalar



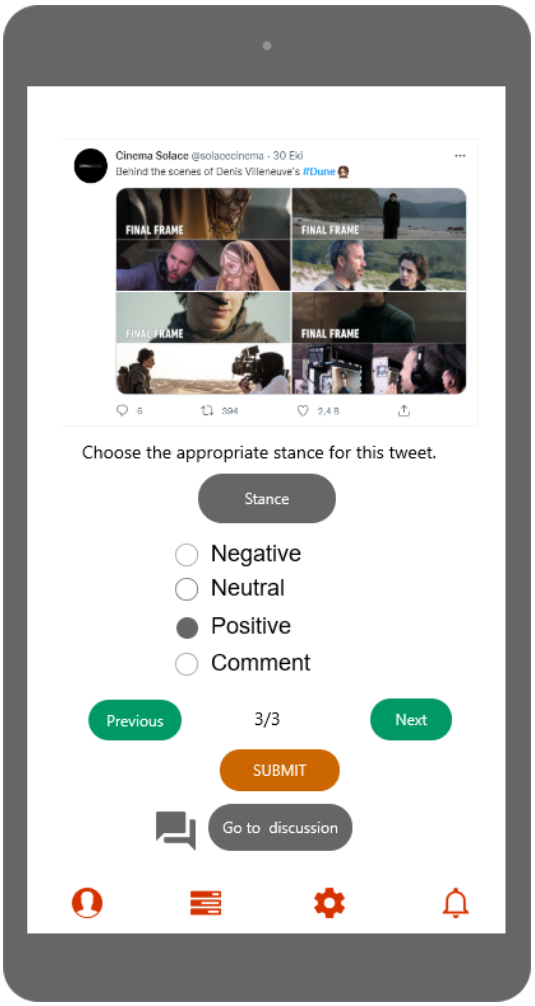
If the post cannot be evaluated based on a scalar metric as in 2.5.5.9, the Client should select from non-scalar options as illustrated in the mock-up above.

2.5.5.11 Voice Chat Mock-up



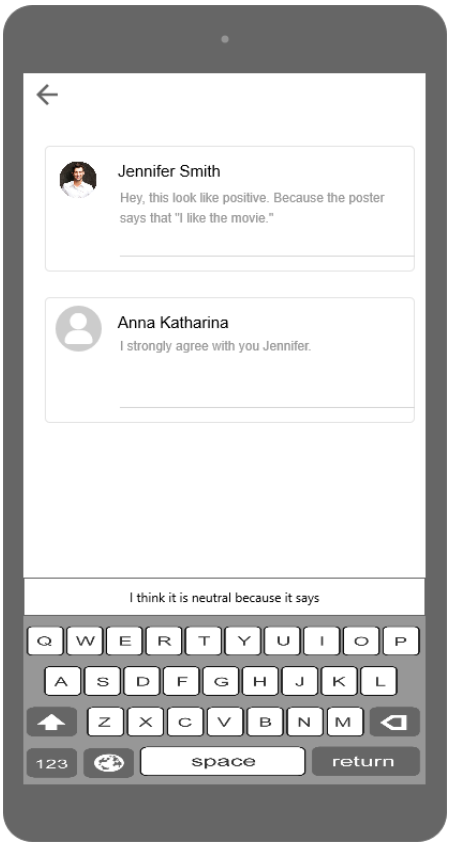
In this page, the Experts can talk about the previous results related to a task that has a high variance by using their microphone. There is a timer on the middle-left of the page. The Experts can talk till the timer is up. When the timer is up or Experts come to an agreement, the Expert makes a new evaluation and submits it to the system.

2.5.5.12 Post with Discussion Thread



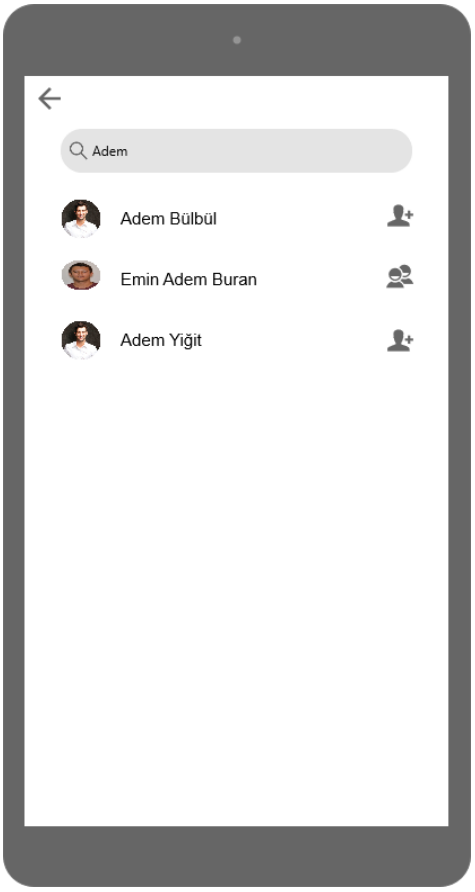
When a Discussion Thread is assigned to the Client, s/he goes to the discussion page by pressing the "Go to discussion" button. S/he cannot submit a new evaluation before contributing to the discussion thread.

2.5.5.13 Discussion Thread Mock-up



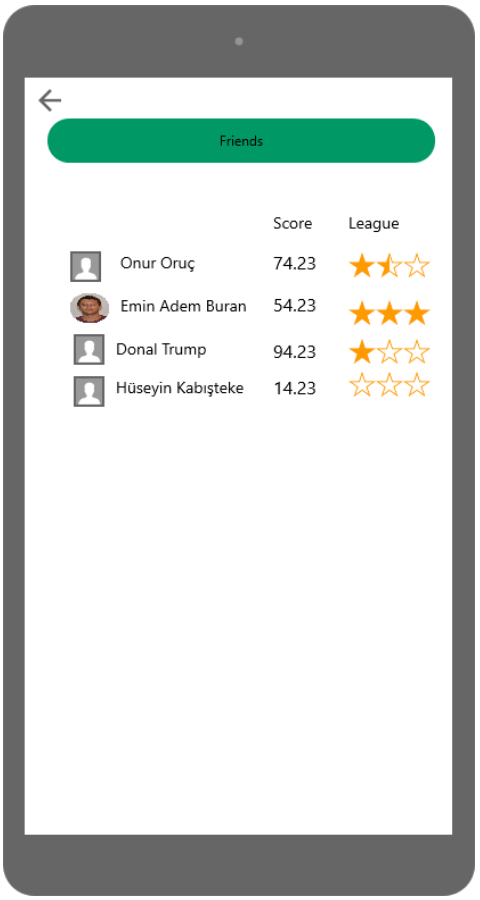
In this page, the Client can view the comments posted for the previously evaluated tasks that have a high variance. The Client may also choose to post a new comment about the results.

2.5.5.14 Find Friend Mock-up



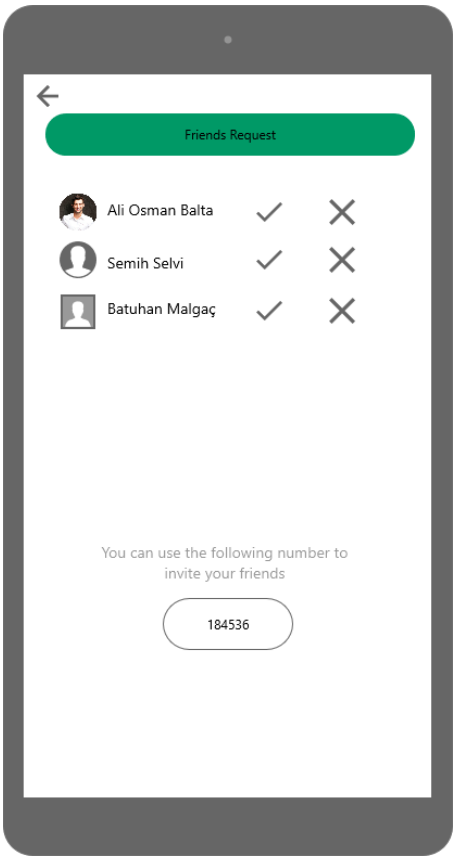
The Client can search for friends by name and all matches are listed on this page. The Client may choose to send a friendship request by simply pressing the button next to the corresponding client's name.

2.5.5.15 Friends Information Mock-up



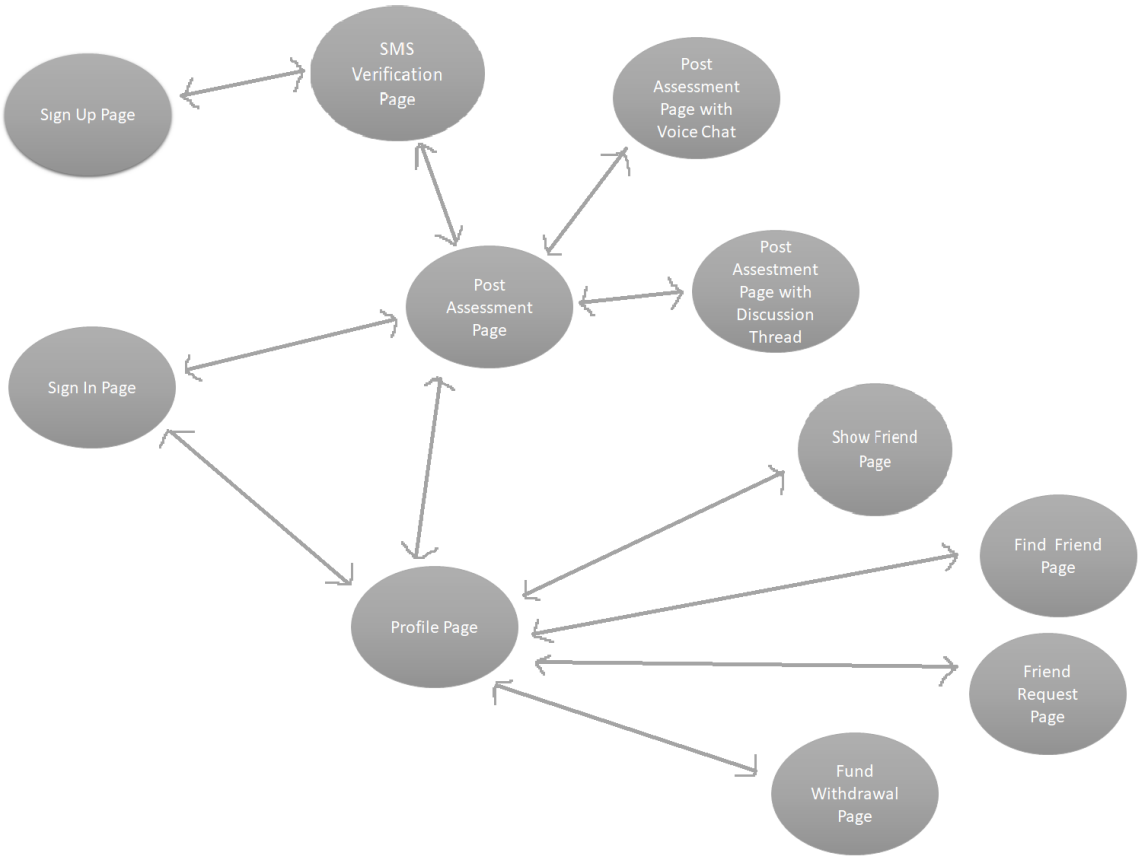
On this page, the Client can see his/her friends' statistics such as their score and league.

2.5.5.16 Friend Requests Mock-up

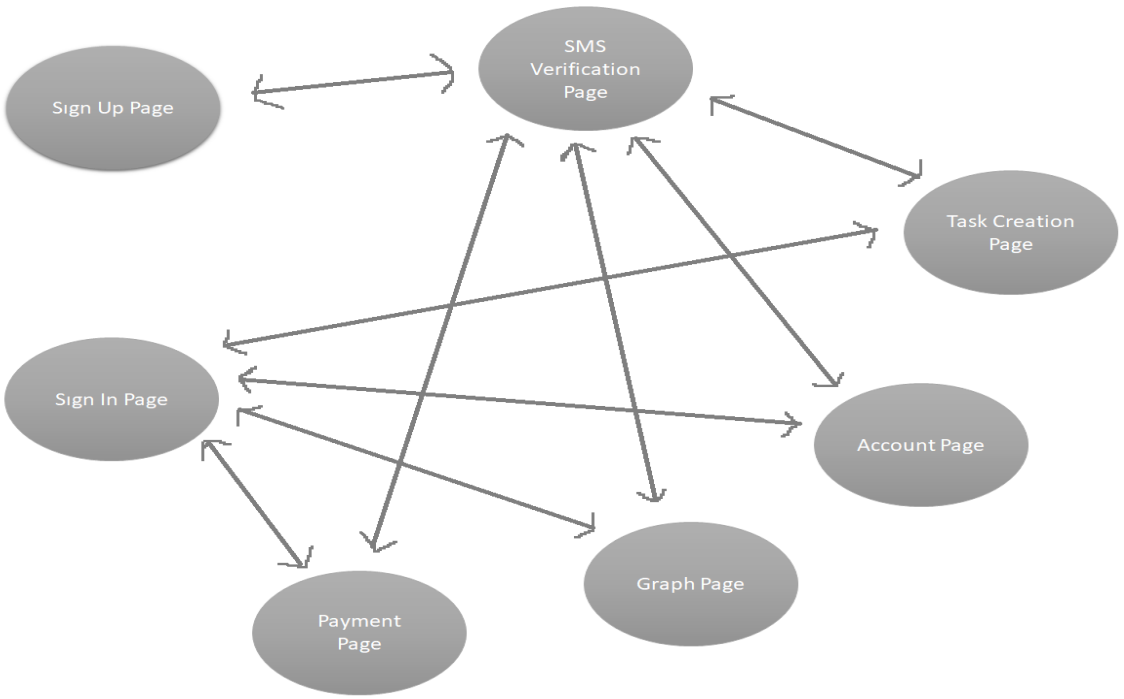


In this page, the Client views the friend requests coming from other Experts. S/he either denies or accepts the reqıests. An invitation code is also displayed to the Expert. His/her friends can sign up to *Laber* with this invitation code.

2.5.5.17 Navigational Path for Mobile App



2.5.5.17 Navigational Path for Web App



3 Other Analysis Elements

3.1 Consideration of Various Factors in Engineering Design

During the analysis phase of our project, we have had to consider several factors relevant to engineering design.

- Public Health: Our approach allows people to work as much as they want when they want. These, at first glance, might sound like positive qualities and they can be. However, they can also have downsides. Extensive use of our application can disrupt a user's work life balance and can be detrimental to their mental and physical health. For this reason, in order to encourage a regular use of our application, we will be providing bonus payment for a predetermined amount of work each day. Excessive work done after the user surpasses this predetermined amount will not yield any bonus payment. We hope that this will discourage users from working in huge chunks of time. We might also consider sending the users messages that recommend taking a break after working for a long period.
- Public Safety: People that work on our application are also likely to work on other platforms that require frequent car-driving such as Uber or DoorDash. Along with that, many of our users will be driving a car daily to go to the places they need. If one of our users decide to work on our application while driving a car at the same time, this will create a safety concern for both the user and the pedestrians and drivers in the area. For this reason, we thought of using GPS to limit the user's maximum speed to be eligible to work on our application. However, with that approach, we prohibit users on busses and trains to work and that would be an unwanted consequence. Instead we decided to make the users sign an agreement form that requires them to not use our application while driving a car or doing any other dangerous activity that requires uninterrupted focus.
- Public Welfare: Our application can provide work opportunities to people, only requiring a mobile phone and internet access as an investment. Therefore, it can be used by people in need as an easy income source. However, there are a couple of considerations regarding this point. Firstly, some people may have access to only a very low end mobile phone with an older operating system. We need to make sure our application can run on such devices if we want to allow these people to work on our

application. For this reason, we need to write efficient code that does not use unnecessary resources. The application should also not be heavy on the battery usage. Secondly, the application will need to exchange data with our servers very frequently. Some users will be paying their internet providers based on the amount of data they download and upload. If our application exchanges any unnecessary data, it might be unprofitable to work on our application for many users. For this reason, we will try to minimize the data exchange via compressing data and not sending/receiving any unnecessary information.

- Global: We plan to allow any expert to work on any project they are eligible for. This means, an expert might be working on a project created by a client from a different country. In this case, different regulations from different regions will need to be taken into consideration.
- Cultural: Since the tasks given to the experts will be very generic, there were not any cultural factors taken into consideration when designing the project.
- Social: During their work sessions, Experts might have to interact with other Experts through our voice chat feature. This might cause problems if some mal-intentioned Experts start harassing others. In order to minimize this risk, we will implement a report functionality and also bans on phone numbers. Since the application requires an SMS confirmation for registration, a ban on a phone number should be able to reliably prevent banned users from using our application.
- Environmental: It is expected that our program will run in many different devices and might use a cryptocurrency for some of the transactions. For energy conservation, it is essential that the program runs efficiently in terms of battery usage, and the selected cryptocurrency is also environmentally friendly. It should be noted that our application will allow Experts to work remotely and this will reduce their carbon footprint by reducing their transportation needs.
- Economic: It is expected that our application will require a big number of small transactions. These transactions may be from across the border which may increase their cost. In order to reduce the number of transactions, we will set a minimum amount that can be withdrawn from an Expert's account. Also, alternative to institutions that provide transaction services like banks, we will provide an option to get paid in a cryptocurrency that our platform supports. In the case that this feature gets implemented in the final product, it will be very important to select a low fee currency that does not have a big environmental impact.

	Effect Level	Effect
Public Health	6	Daily bonuses for a limited amount of work, Warning messages for prolonged work sessions
Public Safety	7	Safe usage agreement form upon registration, Warning message when high speeds detected while using our application
Public Welfare	7	Program needs to run on low end devices, Low battery usage, Low internet usage through data compression etc.
Global Factors	3	Different regulations need to be taken into consideration for different regional versions of the application
Cultural Factors	0	None
Social Factors	7	Report functionality, SMS confirmation on registration and ban on phone numbers
Environmental Factors	8	Efficient code, Environmentally friendly transaction alternatives
Economic Factors	8	Limit on minimum transaction amount, Low fee transaction alternatives

Table 1: Factors that can affect analysis and design

3.2 Risks and Alternatives

We have envisioned five risks we could encounter in our project and come up with alternatives for these risks.

- Risk I: There is a risk of Experts using our application while they are performing dangerous tasks in real life. For instance, they can attempt to complete a given task while they are driving. To handle this risk, we will be presenting every Expert with a warning message/contract, and get them to accept that they will show no such behavior.

- Risk II: There is a risk of certain Experts trying to take advantage of our program by completing tasks rapidly and carelessly in an attempt to make as much money as possible. In order to handle this risk, we have proposed a plan where we cross check the Experts with each other, compute the reliability of each Expert based on the quality of their work, and pay Experts with low reliability less. This should encourage the Experts to complete the tasks carefully and keep their reliability scores high.
- Risk III: Our mobile application will need Internet connection to function. Thus, there is a risk that Experts with poor Internet connection might not be able to load new tasks, or even lose the data corresponding to their completed tasks. To handle this risk, we can keep the tasks given to an Expert in a queue, so that they can keep doing tasks even if they lose connection. Plus, we can store the tasks they complete on their phone until they get reconnected so that they do not lose their newly completed work. Hence, with the plan we propose, Experts with poor Internet connection should not experience any difficulty using our application.
- Risk IV: Given that our program will be using the bandwidth of the Experts, there is a risk that they have to spend too much data and run out of quota. To prevent this, we will be sending as little data as possible.
- Risk V: There is a risk of certain Experts taking advantage of our voice chat functionality and possibly disturb or harass other Experts in the voice chat. In order to handle this risk, we will be adding a "report" functionality so that such Experts can be reported. Thereafter, since we will be documenting each Expert's phone number as a part of their account, we will be preventing these Experts from signing up with different accounts.

Risks	Likelihood	Effect on the Project	B Plan Summary
Risk I	Very Likely	Experts may endanger themselves while using our program	We will present every Expert with a contract so that they can confirm they will do no such thing
Risk II	Likely	The performance of Experts and thus the quality of their work can decrease	We will calculate a reliability score for each Expert and pay them accordingly
Risk III	Somewhat Likely	Experts with poor Internet connection might not be able to load new tasks or lose their completed tasks	We can keep the given tasks in a queue and store the completed tasks in Experts' phones temporarily

Risk IV	Somewhat Likely	Experts may fill their internet quota	We will send as little data as possible
Risk V	Not Very Likely	Some Experts may be harassed while using our application	We will add a “report” functionality and store each Expert’s phone number

Table 2: Risks

3.3 Project Plan

WP#	Work Package Title	Leader	Members Involved
WP1	Mobile App	Hakan	Melisa, Yiğit
WP2	Voice Chat	Onur	Yiğit, Hakan
WP3	Server & Distribution System	Yiğit	Melisa, Hakan, Emin Adem
WP4	Database	Yiğit	Hakan, Emin Adem
WP5	Website	Emin Adem	Onur, Hakan
WP6	Reports	Melisa	Onur, Emin Adem

Table 3: List of work packages

WP 1: *Mobile App*			
Start date: *08.11.2021* **End date:** 20.03.2022			
Leader:	*Mustafa Hakan Kara*	**Members involved:**	*Melisa Taşpınar* *Yiğit Gürses*

Objectives: *This work package aims to implement a fully functional mobile app that can interact with the server. The UX design may go through several iterations. The server interaction behavior should be in accordance with OOP standards and not dependant on back-end implementation.*

Tasks:

***Task 1.1 Implementation of UI & Navigation*:** *Providing a mobile application with a decent user experience.*

***Task 1.2 Implementation of Entity & Control Objects*:** *Providing a mobile application that fulfills our functional requirements.*

***Task 1.3 Implementation of Server Interactions*:** *Providing integration with the back-end.*

Deliverables

D1.1: *Cross-platform mobile application*

WP 2: *Voice Chat*

Start date: *22.12.2021* **End date:** *02.05.2022*

Leader:	*Onur Oruç*	**Members involved:**	*Yiğit Gürses* *Mustafa Hakan Kara*

Objectives: *The purpose is to provide a voice-chat feature that can be used by the mobile application. The voice chat should work in real time without any delays. It should be easily usable by the mobile application by importing the corresponding library.*

Tasks:

***Task 2.1 Voice Chat Library for Android Platforms*:** *Providing a library for voice-chat to be used on the Android platform.*

***Task 2.2 Voice Chat Library for IOS Platforms*:** *Providing a library for voice-chat to be used on the IOS platform.*

Deliverables

D2.1: *A voice chat library*

WP 3: *Server & Distribution System*

Start date: *08.12.2021* **End date:** *20.04.2022*

Leader:	*Yiğit Gürses*	**Members involved:**	*Melisa Taşpınar* *Mustafa Hakan Kara* *Emin Adem Buran*

Objectives: *Providing the back-end infrastructure that can scrape samples from social media and distribute the corresponding tasks to active Experts.*

Tasks:

Task 3.1 Implementation of Web Scraper: *Implementing a web scraper that will scrape social media posts that contain given keywords from the social media platforms selected.*

Task 3.2 Implementation of Distribution System**:** *Implementing a system that distributes the tasks at hand to Experts in a balanced manner.*

Deliverables

D3.1: *Social media post scraper*

D3.2: *Distribution System*

WP 4: *Database*

Start date: *08.12.2021* **End date:** *20.04.2022*

Leader:	*Yiğit Gürses*	**Members involved:**	*Mustafa Hakan Kara* *Emin Adem Buran*

Objectives: *Implementing a database system to store all necessary data and respond to queries. It should minimize duplicate data. Furthermore, it should be implemented in accordance with database design principles and the queries should be efficient.*

Tasks:

Task 4.1 Implementation of Database System: *Implementing a database system which will store our data and respond to given queries.*

Task 4.2 Filling the Database with Mock Data**:** *This mock data will be used for testing the integration with other systems.*

Deliverables

D4.1: *Database System*

D4.2: *Mock Data*

WP 5: *Website*

Start date: *08.11.2021* **End date:** 20.03.2022

Leader:	*Emin Adem Buran*	**Members involved:**	*Onur Oruç* *Mustafa Hakan Kara*

Objectives: *Providing a website where Clients can create new Projects and view the statistics regarding their existing Projects.*

Tasks:

Task 5.1 Implementation of UI & Navigation**:** *Providing a website with a decent user experience.*

Task 5.2 Implementation of Entity & Control Objects**:** *Providing a website that fulfills our functional requirements.*

Task 5.3 Implementation of Server Interactions**:** *Providing integration with the back-end.*

Deliverables

D5.1: *Website*

WP 6: *Reports*			
Start date: *20.08.2021* **End date:** *10.06.2022*			
Leader:	*Melisa Taşpınar*	**Members involved:**	*Onur Oruç* *Emin Adem Buran*
Objectives: *To understand the requirements of each report, have group discussions where we analyze & design the project and write the required reports.*			
Tasks: ***Task 6.1 Project Specification Report:*** *Get an understanding of what is required in the report, analyze our project specifications such as functional requirements, divide the work and write the report.* ***Task 6.2 Analysis Report:*** *Analyze the project, think of scenarios, draw diagrams, consider other analysis factors, write the analysis report.* ***Task 6.3 High-Level Design Report:*** *Create the high-level design of our project with discussions, analyze the current project, and describe these in the high-level design report.* ***Task 6.4 Low-Level Design Report:*** *Create the low-level design of our project with discussions, analyze the current project, and write the required report.* ***Task 6.5 Final Report:*** *Understand the requirements of this report, and write the report accordingly, summarizing our project thus far.*			
Deliverables ***D6.1:*** *Project Specification Report* ***D6.2:*** *Analysis Report* ***D6.3:*** *High-Level Design Report* ***D6.4:*** *Low-Level Design Report* ***D6.5:*** *Final Report*			

Table 4: Description of work packages

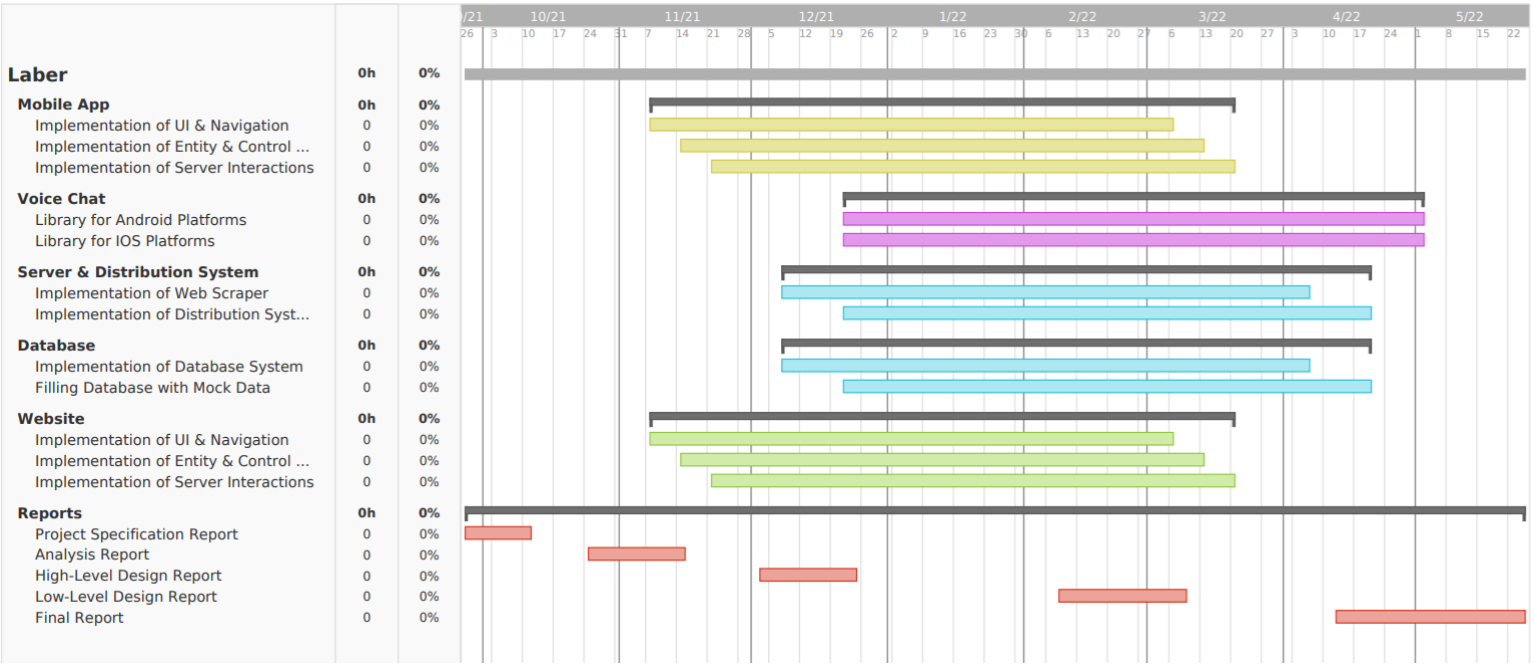


Figure 11: Gantt chart of our project

3.4 Ensuring Proper Teamwork

In our group, each member was assigned a leadership role so as to work in a collaborative environment. How we shared roles within the group can be seen below.

Member	Role
Emin Adem Buran	Checking the work that needs to be done every week according to the project plan and sharing this in the weekly meetings with teammates.
Melisa Taşpınar	Arranging a group meeting every week according to available time of group members.
Mustafa Hakan Kara	Managing the project and checking on work done every week.
Onur Oruç	Researching the existing technologies about how the implementation of the project can be done and sharing information about it with group friends at weekly meetings.
Yiğit Gürses	Arranging meetings, keeping in touch with external factors such as innovation experts, sharing this information with members at meetings.

Table 5: Roles of each member on ensuring proper teamwork

This distribution of roles helped us to make our weekly plans and stick to the project plan. It should also be noted that the tasks we shared here are not the responsibility of only one team member, and all team members share the responsibilities of the tasks written here. The purpose behind task sharing is to ensure that the tasks listed here are done in detail by a group member each week. Namely, even though assignments are made among group members, all team members help each other for weekly tasks.

While preparing project reports, we preferred online tools where everyone can work simultaneously. One of these tools is Google Drive. We made our progress on the project reports via Google Drive and we could check the parts that each other was working on. Thus, we completed the reports with a harmonious study among the group members. We used Visual Paradigm while drawing the diagrams related to the project. Thanks to the Visual Paradigm, we worked together on complex diagrams such as class diagrams, which would create a good skeleton of the project.

We chose to use GitHub to store the code-based work we made in our project. Thanks to GitHub, we worked together on the project website and designed the site together. In later stages, we can work more harmoniously with each other during project implementation by using GitHub.

3.5 Ethics and Professional Responsibilities

Our project includes processing data. As the data may be sensitive, one of our ethical issues is to protect the data and ensure data privacy so that our clients may feel secure while using Laber. To ensure data privacy, we will not use the data without users' details.

Malicious experts may abuse the labeling process by making an agreement to label the post incorrectly in a way that results in significant errors in the task. Our responsibility is to force requirements to prevent this from happening. One of the precautions to prevent this is to validate the experts' expertise level. Moreover, occasionally, the experts will be asked to label posts that we already have accurate labels related to. If they mislabel a significant amount of these posts, their score will be reduced severely and they will not be able to label posts after some reduction in their scores.

3.6 Planning for New Knowledge and Learning Strategies

While analyzing our project, we have also considered what new knowledge we may need to realize it and how we can acquire that knowledge. In order to make our project a valuable learning experience, we have tried to venture into topics and make use of tools that we are not already familiar with. In other words, turning our project into an opportunity for learning was itself one of our considerations during the analysis phase.

For instance, as mentioned before, we have decided to use the React Native framework for our mobile application. This will allow us to create an application that can be available on multiple platforms, while also making use of React's rich libraries. However, it will also be a great learning experience for all members as none of us has had any substantial experience using React Native. We plan on acquiring the knowledge we need on this topic with online learning and some hands-on experience, i.e. learning by doing.

Furthermore, our project includes a server side, with which we are all a bit unfamiliar. In order to acquire the knowledge needed on this topic, we plan on making use of several learning strategies such as online learning using documentations and tutorials, learning from peers that are knowledgeable on the subject, and of course learning by doing.

All in all, our project will likely be an ideal learning experience as it will have several distinct sub-tasks and parts such as a mobile application, a website, a database, a server side, etc. To get the knowledge we will be needing along the way, we will be using online resources, learning from each other, learning by experimenting, getting help by people knowledgeable in the field; noting that these will correspond to learning strategies such as online learning, learning from peers, learning by doing and interviewing experts.

4 Glossary

- Client: Companies and institutions that will be specifying tasks to be completed by Experts using our website.
- Expert: Human experts that will be evaluating social media posts based on specified labels using our mobile application.
- Label: A metric based on which Experts will be evaluating social media posts. These will be specified by Clients.
- Project: Set of specifications, set by a Client, that determines how and when the tasks will be created and distributed.
- Task: A unit of work to be completed by an Expert.
- Voice Chat: A service that allows Experts to discuss in a group voice call, in real time, regarding a task.

5 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] “React Native.” *React Native,* https://reactnative.dev/. Accessed 7 Oct. 2021.