



Training Management App Based on Recommender System



Graduation Project (2)

A project submitted in partial fulfilment of the requirements
for the degree of Bachelor of Science (Computer Science)

2nd Semester -Academic Year 1441/1442 (2020/2021)

Abstract

Many talented people are willing to make an income out of their talents. Either by teaching them or performing them as a service, they sometimes face difficulties finding an appropriate platform that gives them the options of working on the Internet or face to face, self-marketing features and most importantly is to be expanded to include all kinds of talents. People who are interested in improving these talents are also struggling to find the perfect match trainer who can fulfill their needs with a reasonable price and satisfying professionalism.

The intelligent application is designed using React Native to host qualified trainers and recommend them to people who need to improve their talents based on a *content-based* recommender system that uses the user's data, the nearest location, and the highest rating to make predictions. The platform is also considered as a social media application that allows trainers to advertise themselves using profile pages and posts, and that improves the user experience as well. It also includes a lot of motivational activities for trainees to have the best outcomes.

The project's primary purpose is to examine how the recommender system enhances the reciprocal process of matching the most suitable talents trainers to the interested trainees on a custom application. The idea is unique and wanted in the light of Saudi Arabia's 2030 vision that encourages local talents.

Keywords: Recommender System; Training; Trainee; Trainer; Talent

Acknowledgement

In the Name of Allah, the Beneficent and the Merciful

First of all, we would like to thank God for all the blessings He has given us.

Besides, we are thankful to Dr.Madawi for her patience and kindness during the semester.

We also thank Taibah University for what it has provided us for all these years.

Finally, to our family for being with us

Table of Contents

Abstract.....	i
Acknowledgement.....	ii
List of Figures.....	v
List of Tables	vi
List of Abbreviations	vii
Chapter 1: Introduction	1
1.1 Introduction	1
1.2 Problem Definition	2
1.3 Project Objectives.....	3
1.4 Project Scope.....	3
1.5 Project Timeline.....	3
1.6 Judan	4
1.7 Document Organization.....	5
Chapter 2: Literature Review.....	6
2.1 Background.....	6
2.1.1 The Recommender System	6
2.1.2 Recommender System Approaches.....	7
2.2 Related Work.....	13
2.2.1 Similar Mobile Applications:.....	13
2.2.2 Our Proposal Versus Previous Work	17
2.3 Summary	18
Chapter 3: System Analysis	19
3.1 Survey Analysis.....	19
3.2 Requirements Elicitation	25
3.2.1 Functional Requirement	25
3.2.2 Non-Functional Requirements	26
3.2.3 User Requirements or Domain Requirements	27
3.3 Requirements Specification	28
3.4 Developmental (or Research) Methodology.....	37
3.5 Summary	40
Chapter 4: System Design	41

Chapter 1: Introduction

4.1	Architectural Design	41
4.2	Object Oriented Design.....	42
4.2.1	Structural Static Models	42
4.2.2	Dynamic Models	43
4.3	Data Modeling.....	46
4.4	User Interface Design.....	48
4.5	Summary	57
Chapter 5: System Implementation		58
5.1	Introduction	58
5.2	Tools and Languages.....	58
5.2.1	React Native.	58
5.2.2	Node.js	61
5.2.3	Application Programming Interface (API).	61
5.2.4	Android Studio.....	61
5.2.5	Visual Studio Code (VS)	62
5.2.5	Structured Query Language (SQL)	62
5.3	Mapping Design to Implementation	62
5.4	Main/Most Important Codes	64
5.3.1	Request Page code	64
5.3.2	The weighted average rating	66
5.3.3	Recommender System Code	66
5.3.4	Authentication.....	69
5.5	System Testing	70
5.6	Results and Discussion	74
5.7	Summary	74
Chapter 6: Conclusion and Future Work.....		75
6.1	Conclusion.....	75
6.2	Goals Achieved	75
6.3	Limitations and Future Work	76
6.4	Our Contribution:	76

List of Figures

Figure 1.1 Survey results of the importance of having hobbies [1].....	1
Figure 2.1 Recommender system approaches.....	7
Figure 2.2 Content based approaches.	7
Figure 2.3 Pearson correlation algorithm.....	9
Figure 2.4 TutHive application logo	14
Figure 2.5 QuickTutor application logo.....	15
Figure 2.6 HalaPro application logo.....	16
Figure 2.7 Gooroo application logo.....	17
Figure 3.1 Age groups of the participants.....	19
Figure 3.2 Potential trainers' and their expectancy of facing self-marketing difficulties.	20
Figure 3.3 Participants' recognition of the existence of unqualified trainers.	21
Figure 3.4 The impact of having a personal trainer in the speed of improving a hobby. ..	22
Figure 3.5 The confidence in reaching an agreement through social media.	22
Figure 3.6 The difficulty of finding a close trainer.....	23
Figure 3.7 Lacking the encouragement needed for the improvement of hobbies.....	23
Figure 3.8 The support of Saudi talents.....	24
Figure 3.9 Use case diagram.....	29
Figure 3.10 Trainer use case	30
Figure 3.11 Trainee use case.....	30
Figure 3.12 Admin use case.....	31
Figure 3.13 The phases of development model [18].....	38
Figure 3.14 The phases of Waterfall model [18].	38
Figure 4.1 Architectural design.....	41
Figure 4.2 The class diagram.....	43
Figure 4.3 Activity diagram.....	44
Figure 4.4 Recommender sequence diagram.	45
Figure 4.5 The sequence diagram.....	46
Figure 4.6 The entity relationship diagram.....	47
Figure 5.1 The illustrate the flow of flux.....	59
Figure 5.2 Payment interfaces.....	63
Figure 5.3 Trainer's navigator.	64
Figure 5.4 Map start point based on trainer's location	65
Figure 5.5 Convert latitude and longitude to kilometer	65
Figure 5.6 The weighted average rating	66
Figure 5.7 Recommender System Algorithm	67
Figure 5.8 Recommender System front end.	68
Figure 5.9 Recommender System back end.....	68
Figure 5.10 Facebook Authentication.....	69
Figure 5.11 Encryption Authentication.....	70

List of Tables

Table 1.1 Shows the time period required of each task for the first semester.....	4
Table 1.2 Shows the time period required of each task for the second semester.	4
Table 2.1 Recommender system approaches summary [4].	12
Table 2.2 Summarize the differences between training management applications.	18
Table 3.1 Non-functional requirements.	27
Table 3.2 Create account use case.	31
Table 3.3 Search use case.	32
Table 3.4 Request trainer use case.	32
Table 3.5 Give feedback use case.	33
Table 3.6 View progress page use case.	33
Table 3.7 View recommendations.	34
Table 3.8 Chat use case.....	34
Table 3.9 Edit profile use case.....	35
Table 3.10 Manage posting use case.....	35
Table 3.11 Give evaluation use case.....	36
Table 3.12 Delete violation use case.....	36
Table 3.13 Confirm Payment use case.....	37
Table 3.14 Tools and technologies.	40
Table 5.1 Comparison of all the four platforms.....	60
Table 5.2 System Testing.....	71
Table 5.3 System Testing2.....	74

List of Abbreviations

Abbreviations	Definition
IOS	IPhone Operating System
COVID-19	Coronavirus Disease 2019
ER	Entity-Relationship Diagram
RS	Recommender System
CF	Collaborative Filtering Technique
BN	Bayesian networks
NB	Naïve Bayes
HF	Hybrid Filtering Technique
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JSX	JavaScript XML
API	Application Programming Interface
UML	Unified Modeling Language
UI	User Interfaces
UX	User Experience
AES	Advanced Encryption Standard
API	Application Programming Interface

Chapter 1: Introduction

1.1 Introduction

Talents and hobbies have a significant role in human's life. Talent is the skillfulness and expertise that a person has extraordinarily, while a hobby is defined as a joyful interest practiced in free times. Talents and hobbies need to be practiced continuously for constant improvement. They add many benefits to our lives, including organizing and managing time, growing a career, bonding with others, creating new friendships, and gaining confidence. In January 2016, a survey about hobbies was shared by AYTM among 18 - 65+ years old adults in The United States to show the importance of hobbies. The survey got 400 responses in one day. Figure 1.1 shows the results of answering the question: "Do you think having hobbies is important?"

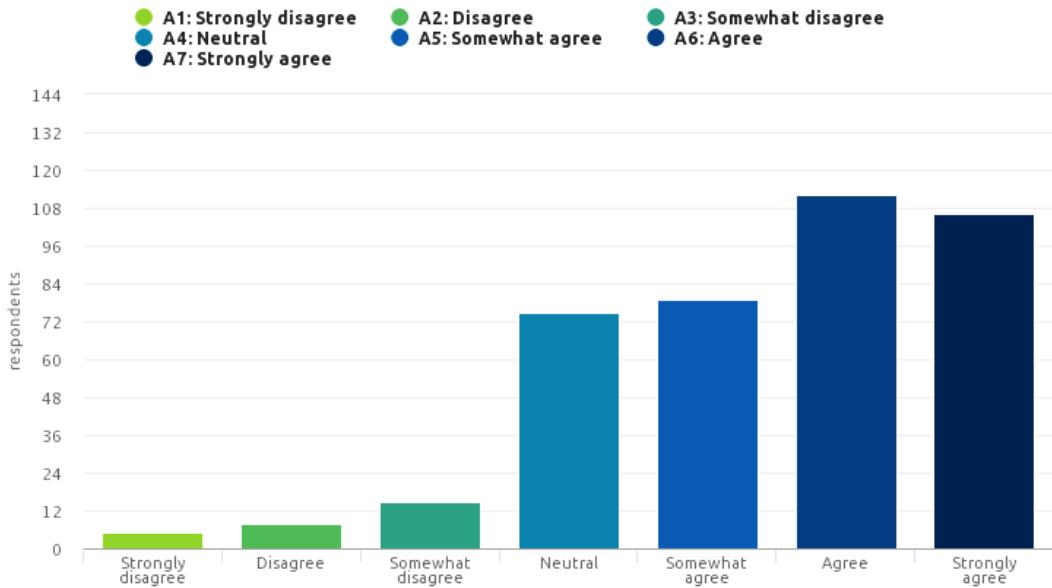


Figure 1.1 Survey results of the importance of having hobbies [1].

One of the aims of The Kingdom of Saudi Arabia 2030 vision is to support the local talents and skills, and that is what our work is going to help with. The primary project purpose is to help trainees find the best match to fulfill their needs using specific information and advanced recommender system mechanisms.

In this work, we aim to improve the training process of hobbies/talents using the most professional and effective techniques and provide the potential for talented people to make the best benefit out of their talents. This work will provide all the features and facilities to make an encouraging, joyful, and educational environment for both talents trainers and trainees. The project is a mobile app (for both IOS and Android) developed using the React Native framework.

1.2 Problem Definition

From the divergent fields of talents, everyone either has a unique talent to teach or an interesting hobby to learn. Most of the time, people need an expert to pass that specific knowledge, theoretically or practically. However, most of the available platforms are either concrete for only one field like singing for example, or not specific at all, like using Instagram for that purpose. In both options, the choices are minimal and can end up with bad results like choosing a non-qualified tutor or not finding one at all, especially in case of specific fields and rare talents. Equally important is the existence problem of using an untrusted training platform. The process of finding the ideal trainer for the trainee has many difficulties. We highlighted the main issues related to our topic as following:

- The inability of finding trainers especially in case of rare talents.
- The place of training may be hard to reach or risky due to the pandemic situations (COVID-19).
- The shortage of nearby trainers to meet face-to-face.
- The disqualification of some trainers
- The high salary of trainers in most cases.
- The lack of the encouragement needed to support the talent or hobby.
- The fear of choosing and meeting an untrusted trainer.

The same goes for the people who wish to create an income out of their talents as trainers and easily reach people who want their help. They usually need a place which provides the facilities for self-marketing and the ability to attract interested clients to teach their skills or maybe have the chance to provide them as a service.

One of the main issues in our work is the complex process of choosing the most qualified trainer with the nearest location and the same interests. The process includes many factors that would take a long time and considerable effort.

1.3 Project Objectives

In this project, we intend to study and build a training management app based on the recommender system. We define our objectives as follows:

- To study the mathematics and algorithms of the recommender system and distinguish the differences between its approaches.
- To identify which approach gives the closest match according to what our system needs.
- To develop a mobile application that helps managing the training process and helps facilitating the complexity of choosing the ideal trainer by giving the best recommendations.

1.4 Project Scope

The study will focus on the talents in Saudi Arabia, especially the western region. It targets talented people who are willing to make an income out of their talents by either performing them as a service or teaching them as trainers (online or face to face), and targets interested people who are willing to improve their talents with a trustful expert.

It will only focus as a start on the talents that can be performed on the stage as an entertaining show, and that includes (instruments playing, singing, dancing, drawing, magic tricks, acting, authoring and diction, acrobatic and sports, stand-up comedy, and science shows).

1.5 Project Timeline

The project timeline is a method aimed at scheduling tasks to ensure that they are completed on time. It also helps the team members to manage and organize the project in an efficient manner. The table 1.1 shows task periods for the first semester.

Chapter 1: Introduction

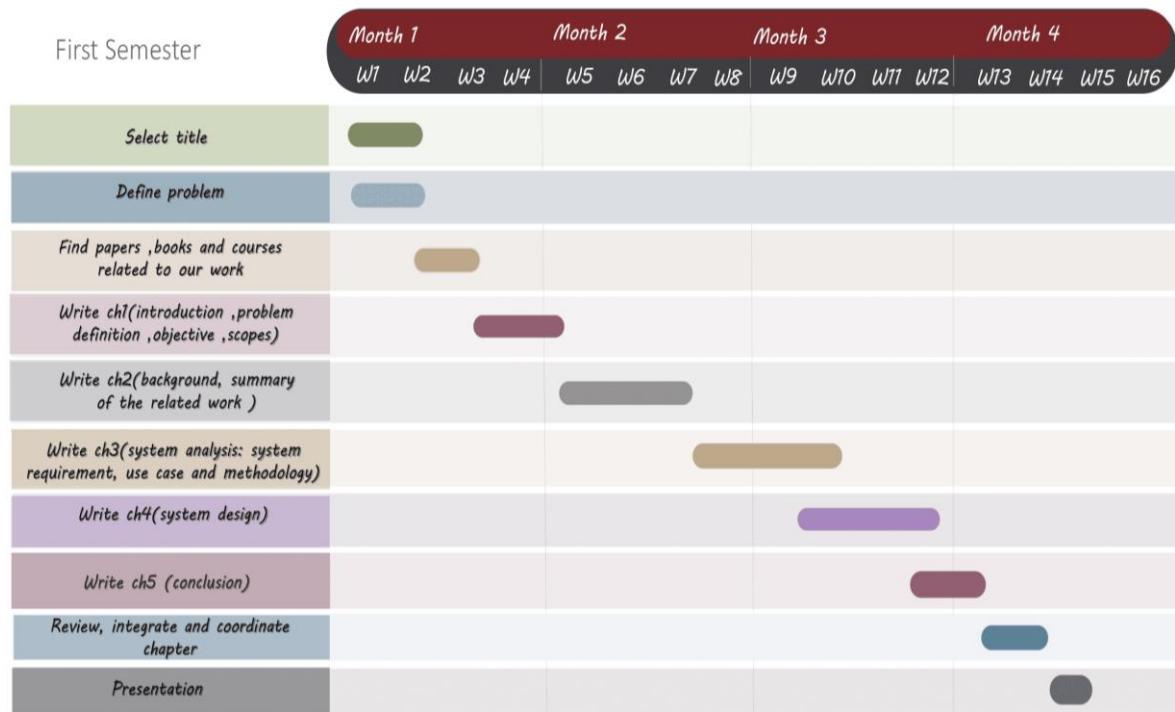


Table 1.1 Shows the time period required of each task for the first semester.

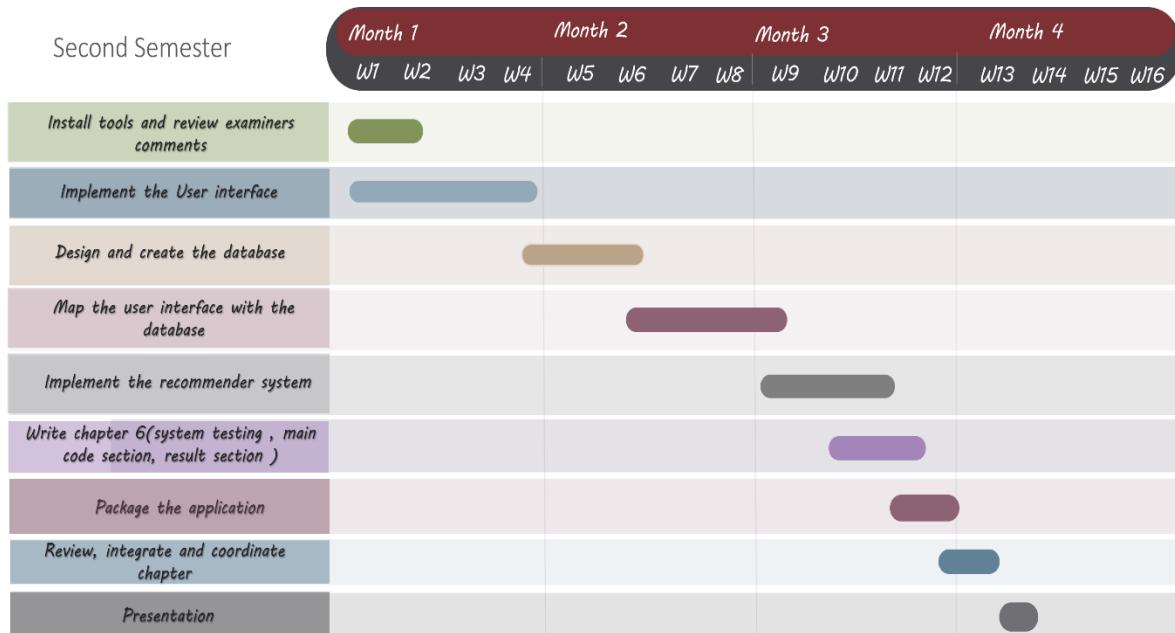


Table 1.2 Shows the time period required of each task for the second semester.

1.6 Judan

Judan is the name of the highest blackbelt level (the 10th-degree black belt in Dan rank). The progress reflected in the different belts in Karate is depicted in our application through reaching a high level of professionalism at the end.

1.7 Document Organization

This project consists of five chapters. These chapters are organized to reflect the scientific steps toward our main objective, and to show the main phases of developing our application (Judan). A brief description about the contents of each chapter is given in the following paragraphs:

Chapter 1 introduces the main terminologies, explains the problem definition, determines project objectives, sets the timeline for each chapter, determines the scope of the work, and shows the project layout.

Chapter 2 provides the reader with an overview of the literature review, sharpens the knowledge needed as a background describing the approaches and techniques needed in detail, and shows a number of related applications that share common background with Judan application and compares them against others.

Chapter 3 focuses on the survey analysis, requirements elicitation, and requirements specification using use case diagram and its tables for the different actors (trainer, trainee, admin).

Chapter 4 clarifies different design aspects, including the architectural design, the object-oriented design for both static view (class diagram) and dynamic view (activity diagram, sequence diagram), database design (ER diagram), and the user interface design.

Chapter 5 focuses on the implementation process and includes the main tools and code sections that we used to build Judan application. The chapter also describes how we mapped the design described in chapter 4 and the changes we made. Finally, it shows the success of the testing phase and its coverage.

Chapter 6 summarizes the conclusion of this work, the goals achieved, and limitations.

Chapter 2: Literature Review

In this chapter, we will cover a set of valuable techniques and similar proposals in the context of this work. In section 2.1, we will overview the background knowledge about the recommender system and its approaches. Section 2.2 is divided into two sections. First, we will discuss existing applications that share a common background regarding talent training. The second section compares the similarities and differences between our work and other applications. Section 2.3 will summarize the main points covered in this chapter.

2.1 Background

2.1.1 The Recommender System

The recommender system (RS) is known as the techniques and software tools used to provide suggestions for the item that might interest a specific user. The term "item" refers to the general term for which the system recommends to a user. Usually, the recommender system focuses on one type of item (such as news or movies) [2].

When e-commerce sites began to expand, the recommender systems started to become popular. This system aims to provide the best suggestions to users from sets of options offered by the sites that might go wrong without using it. Application development tends to add a recommender system to shorten user's searching time. This technology has been used in all large companies such as Netflix, Amazon, Facebook, etc.

The recommender system predicts the user's interest. This prediction can be divided into personalized and non-personalized recommendations. The first type depends on the user's experience which can be done using the user ratings. This technique is known as *collaborative filtering*. Non-Personalized suggestions usually are easier to make and are generally published in newspapers or magazines. The Non-personalized recommendation system recommends products to users based on the average of other users' opinions. In this type, the recommendations are not based on the users, so the same recommendations are given for all users [3].

Data mining is the computational process of finding patterns and extracting information from significant amounts of data. Web mining is further divided into three categories: web structure mining, web content mining, and web usage mining. The

Recommendation system is considered web usage mining in which it discovers interesting usage patterns to understand user's behavior [4]. Data collection is the main fundamental and the backbone of the recommender system. The gathered data is usually one of these categories: demographic data, production data, user behavior, and user rating. Demographic data is the information required in registration (sign up), including country, gender, age, hobbies, and more. Production data are the categories or classifications that merchants add to their items to make it easier to appear in the user's search like the tags under YouTube videos. User behavior is achieved when the system monitors the user's activity like the number of clicks, the time duration of browsing, and the first viewed items. User rating is a user's ability to interact with what they experienced by giving their opinions as rating inputs [5].

2.1.2 Recommender System Approaches

The basic recommender system model works on two different data types, user-item (such as the rating history) and attributes information (such as relevant keywords). Using the two categories, the recommender system can be divided into three main sections: *content-based filtering*, *collaborative filtering* technique, *hybrid filtering* technique, see Figure 2.1 [4].

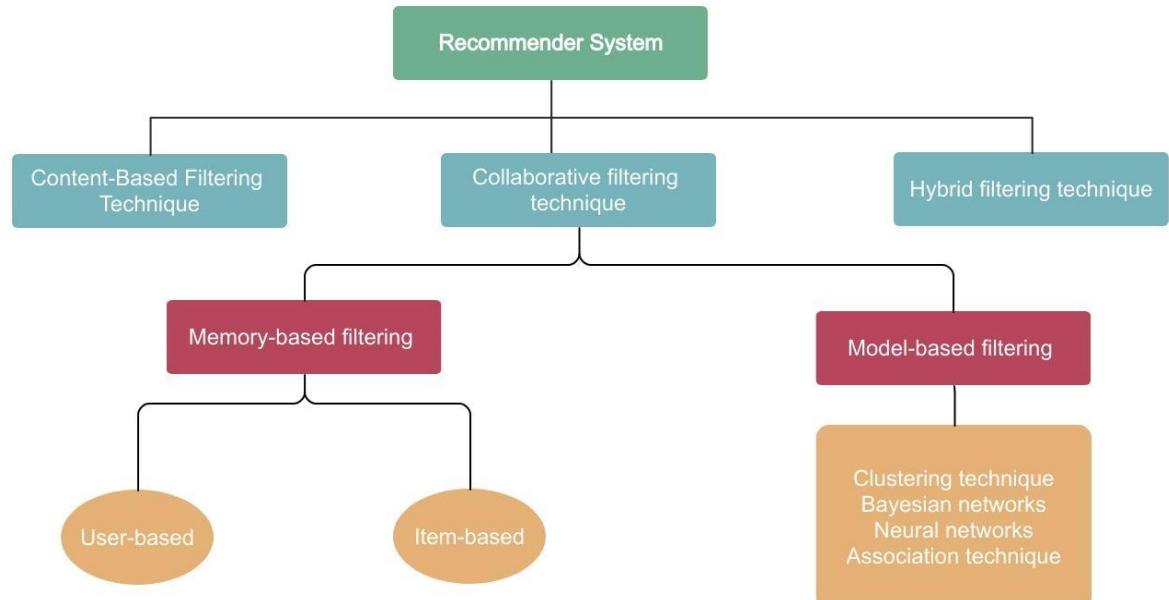


Figure 2.1 Recommender system approaches.

Here, we will discuss each approach in detail:

- **Content-Based Filtering Technique**

Content-based filtering is known as one of the main types of the recommender system. This type is designed based on a comparison between the description (contents) of a set of item attributes and a user profile [4]. A user profile refers to all the information gathered from a user. It means it is a formal reflection of the user's desires, including the user's needs, wishes, and interests [6]. In such cases, user reviews and ratings on other films are enough to discover significant recommendations. The approach depends on the user's interest itself. For example, when a user likes a type such as an action movie, it takes that as key and uses this to predict later. Thus, this technique can be a valuable tool used in intelligence applications.

Content-based filtering connects the user with an item similar to the user's interests. It uses their previous performances or explicit feedback about the item to know the similarity. This similarity does not necessarily mean that the rating is identical to correlations across users, but it may be based on the similarity of item attributes in which the user has liked in the past, unlike the collaborative systems type, which is based on users ratings in addition to target user ratings [7]. This type uses an item-to-item correlation rather than a user-to-item correlation to make recommendations as shown in Figure 2.2. The following steps are carried out in the method of making recommendations:

- Gathering some of the general content data about the item (movie name, Actor, Director, Rating and Year, etc).
- Data analysis and extraction of useful functionality and features regarding its content.

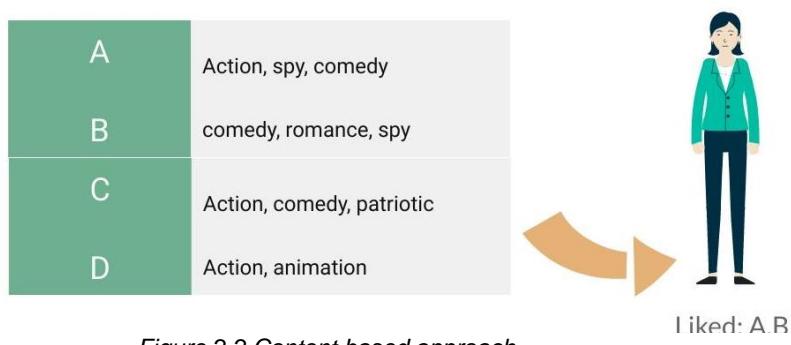


Figure 2.2 Content based approach.

The advantage of the *content-based filtering* approach is that it does not need other users' data and can recommend items to a user with a unique taste. However, items are limited to their initial features or descriptions [3].

- **Collaborative Filtering Technique**

Collaborative filtering is considered the most popular technique in the recommender system [5]. It is an approach of the recommender system where its suggestions depend entirely on the historical information of many contributors and collaborators who have shared preferences. This is the reason why they refer to *collaborative filtering* as "people-to-people correlation" [7]. These collaborations are usually made indirectly by taking their opinions or ratings to know the similarity in taste between users. *Collaborative filtering* is divided into two main types: *model-based filtering* and *memory-based filtering*.

Memory-based filtering calculates similarities between users or items to conclude what the user will like next mathematically. It also has two techniques, *user-based* and *item-based* [4].

User-based collaborative filtering calculates similarities between multiple users based on their historical ratings on a group of similar items and then predict current user's rating by finding the weighted sum of deviations from mean ratings of users who previously rated the same item and the user mean rate as shown in Figure 2.3 [4]:

$$s(a,u) = \frac{\sum_{i=1}^n (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^n (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i=1}^n (r_{u,i} - \bar{r}_u)^2}} \quad p(a,i) = \bar{r}_a + \frac{\sum_{i=1}^n (r_{u,i} - \bar{r}_u) \times s(a,u)}{\sum_{i=1}^n s(a,u)}$$

Figure 2.3 Pearson correlation algorithm

In the first equation, $s(a,u)$ represents the similarity between two users, a and u . $r\{a,i\}$ is the rating which is given by user a to item i , $r\{a\}$ is the average rating given by user a and n is the total number of items in the user item space. For the second equation, $p(a,i)$ is the prediction for the current user a for item i , $s(a,u)$ is the similarity between users a and u , and n is the number of users in the area [4].

Item-based collaborative filtering calculates similarities between multiple items using the user-item matrix to predict a user's preference to an item based on his/her ratings

of similar items. The similarity can be calculated by three methods: cosine-based similarity, correlation-based similarity, or adjusted-cosine similarity [4].

Model-based collaborative filtering begins with developing a model of user rating to provide item recommendations. In this category, algorithms take a probabilistic approach and anticipate a typical filtering process as a computation of the expected value of a user's forecast, considering their ratings in other items. The model is constructed using machine learning or data mining techniques. After the model is created, it will only use this developed model to make recommendations. These techniques can quickly recommend a set of products or items because they use a pre-computed model, and the recommendation results they produce are similar to neighborhood-based recommendation techniques. *Model-based* recommendation systems use small data sets called models. The design of the model is to extract some information from a huge database related to specific parameters or attributes and use this model every time without using a huge database [4]. For *model-based* CF, there are four common approaches such as clustering technique, Bayesian networks, neural networks, and association technique

Clustering technique [8] works based on the assumption that users in the same group have the same interests; therefore, their ratings of products are similar. Thus, users are divided into groups called clusters, which are defined as a group of similar users.

Bayesian belief network has a decision tree on each node. The model has a decision-tree for each node of BNs, one of which corresponds to each item in the domain, and the state of each node corresponds to the possible level of each item [9]. Bayesian networks (BN) are often used for classification tasks. Motivated by the simplicity and accuracy of the naive Bayes classifier (NB), BNs are being used a lot as an example recognition, fault diagnosis, and other classification tasks [10].

Neural Networks is doing the following things: First, it uses a user-item rating matrix to obtain users' and items' characteristics and then treats these characteristics as inputs to the neural network. The output layer will obtain some probability values, which represent the probability of the score that the user may give. In the end, the score with the highest probability will be used as the final prediction result [11].

The task of association analysis is to find patterns in an item set for items that occur together to a specific extent [12]. Searching for association rules can be valuable to any commerce company as it shows evidence for which items that are purchase together.

- **Hybrid Filtering Technique**

Hybrid filtering (HF) combines *content-based filtering* and *collaborative filtering*, which can be more effective in some cases. *Hybrid filtering* can be executed in several ways: by separating *content-based filtering* and *collaborative filtering* predictions and then combining them, or adding *content-based* capabilities to *collaborative filtering* (and vice versa), or unifying techniques into a single model. Some studies empirically compare hybrid performance with pure *collaborative* and *content-based filtering* methods and show that hybrid methods can provide better recommendations than using only one recommendation system. You can also use hybrid filter methods to deal with some common problems found in recommender systems, such as cold start and the rarity problem [4].

Using multiple recommendation techniques can overcome the weaknesses of a single technique in a combined model. A combination of approaches can be performed in any of the following ways: separate implementation of the algorithms and combination of the result, use of some *content filtering* in *collaboration* approach, using some common filtering in the *content-based* approach, creating a unified recommendation system that combines both approaches. It overcomes the problem of rarity and cold start. *Hybrid Filtering* combines the different recommendation methods to obtain better optimization of the system and avoid certain limitations and problems related to the techniques of the pure recommendation system. The main objective Behind *hybrid filtering* is the fact that a combination of algorithms will provide more accurate and effective recommendations than a single algorithm since the disadvantages of one algorithm can be overcome with another algorithm. Using multiple recommendation methods can overcome the weakness of a single technique in a combined model. Hybrid recommendation systems are used to describe any recommendation system that combines multiple recommendation methods together to produce results [4]. All in all, Table 2.1 shows the differences between recommender system approaches.

	Collaborative filtering	Content-Based filtering	Hybrid filtering
Number of users	Recommendation based on many users having similar interest	Recommendation based on single user	Recommendation based on many users having similar interest
Shortcomings	- Scalability - Data sparsity - Cold start problem - Grey sheep problem	- Limited content analysis - Over specialization	- Increased complexity - Increased expense of implementation
Advantage	Serendipitous recommendation	User independence and transparency	Overcome cold start problem and data

Table 2.1 Recommender system approaches summary [4].

In this work, we found it suitable to implement content-based filtering approach. In which the recommendations are focused on a single trainee by finding similarities among trainers who share common fields, interests, and region. When a user gives a trainer high rating, the system will take the information of that trainer as input to calculate the similarities. Trainer's interests are manually selected and managed by the trainer as tag boxes. Furthermore, the region is taken from sign up data when the trainer puts the country and city. The recommendation process also takes the rating into account. When the recommender finds trainers with similar fields and interests but different rating, it will give the priority of showing in recommendations to high rated trainers.

The reason of choosing this approach is mainly because of other approaches disadvantages. Collaborative filtering suffers from cold start problem, which means the ineffectiveness in the beginning of releasing the application because of the shortage of users. Collaborative filtering depends on having many users to calculate the similarities between them. So, it is better to use other approaches as a beginning. In addition, grey sheep is also considered as disadvantage, in which some users have unique and unusual preference, so they do not match or mismatch most users []. Other reason is the suitability of focusing on the content of trainers as they tend to share fields, interests, and of course the region.

2.2 Related Work

A brief review and analysis of the most relevant work that will clarify several similarities, differences, and what makes our work stand out. The references have been selectively chosen based on many similarities between them and our work in terms of the idea and technologies used. The section will be divided into two subsections, one for mobile applications and the other for papers and books.

2.2.1 Similar Mobile Applications:

- **TutHive Application :**

TutHive mobile application was developed by Thamer Al Horani for both iOS and Android platforms [14]. The app is used to educationally help +5 years old people. It only supports the English language, and it is available in Saudi Arabia and many countries.

The application enables users to log in as a tutor or as a student. It provides for students a nearby tutor to teach school or university academic topics; or a trainer to teach a skill, talent, a language, or a fitness program; by manually searching and selecting a category of interest to choose one of the available trainers/tutors in the same country or by viewing the map and directly selecting the nearest one from there. The application finds the nearest tutor/trainer so that students can contact them via the chat section and arrange an appointment. The application can also show the highest rating tutors/trainers in the same city and suggest them under a specific section. Furthermore, the TutHive Application allows users to see the near educational events and register to join them. Moreover, it has the option of providing a private group tutoring for giving better experience.

The application focuses only on finding the nearest tutor/trainer and lacks the option of choosing remotely case of the online classes. The process of automatically recognizing the city is also not precise and can make mistakes. The application depends on the chat section to make an appointment and arrange the training approach with the tutor, which might weaken the user experience. Another critical issue is that TutHive does not have any encouragement methods for supporting users to keep improving. Finally, the recommended approach for

suggesting tutors/trainers depends entirely on the rating and lacks personal preferences.



Figure 2.4 TutHive application logo

- **QuickTutor Application :**

QuickTutor mobile application was developed by QuickTutor LLC for both iOS and Android platforms [15]. It is made for +12 years old users. It supports the English language and 43 more languages including the Arabic, and it is not available in Saudi Arabia.

In this application, the user has the choice to log in as a quick tutor or as a student. After students choose what they are interested in among various subjects, topics, and skills, they can select a quick tutor who fits their needs. The system manually matches trainers to trainers according to the selected category and allows them to contact via the chat section to arrange appointments and other details. Communications can be done by real-time (online) video calls or in-person by customized meet-up menus, based on what the student chooses. The last significant feature in this application is something called “QuickRequest” that allows students to customize a request for the company to manually choose the ideal trainer for them based on the type of training, the subject, the date, and the time.

On the other hand, regarding this application, we found some missing features. The suggestions are random and not helpful in most cases. Additionally, it lacks the encouragement features. The “QuickRequest” feature is not ideal because it is time-consuming. The student has to wait until the company gets his/her request and searches for a suitable tutor/trainer, which can take a long time



Figure 2.5 QuickTutor application logo.

- **HalaPro Application :**

HalaPro mobile application was developed by Mohammed Al Bitar for both iOS and Android platforms [16]. It improves people's life who are +4 years old by providing the facilities for the education and training process. It supports the English and Arabic languages, and it is available in Saudi Arabia and a lot of other countries (almost all the countries).

It provides experts the ability to work independently and tutor/ train in their free time, whenever and wherever suits them. As well as providing the option of singing in as a student. They can find their matching tutor/ trainer manually by entering a subject of interest and selecting a country, and then scrolling among the available results to choose the most relevant one. HalaPro provides the two online and in-person training options, depending on what they have agreed on via chat messages. Furthermore, it has attendance and timetable in addition to document sharing and syllabus.

The application has several missing futures. For example, after choosing a major to be learned, there is a high possibility that the trainees will not find any trainers in the same area (country) to meet them in person. The app has many countries, which makes it hard to obtain trainers for all specialties in every country, and that is why the app mostly focuses on online classes. Furthermore, HalaPro does not have a way to encourage the trainees to make them more interested in the app, and chatting is the only agreement method between tutors/trainers and students to discuss and make a deal. Finally, the filtering process focuses only on the chosen country and the entered subject, which might not be enough especially when the

user wants to choose the most qualified tutor/trainer in all countries to arrange an online class regardless of the country.



Figure 2.6 HalaPro application logo

- **Gooroo Application :**

Gooroo mobile application was developed by BOK Solutions 1nc for iOS platforms [17]. It educationally helps +4 years old people. It only supports the English language, and it is not available in Saudi Arabia.

The application displays many questions when the user requests a trainer to ask about the appropriate way for trainees to learn. With the Gooroo app, the trainee can manage the sessions by using an electronic system specialized for scheduling appointments and choosing the training type rather than using the traditional massages style. Communications for extra details can be done via messages in the chat section. It also has online and in-person training options, and it allows trainees to give feedback after each session.

Gooroo application links the trainer to the trainees through several questions only about preferences instead of having a customized recommender system. It does not take into consideration the combination of personal sign-up data, the rating, the location, and the preferences, which are all part of the recommendation process. Additionally, the recommendation of Gooroo suggests only one trainer rather than giving a list of recommendations for students to choose. Likewise, it uses a pre-payment method, which prevents users from choosing trainers and contacting them unless they subscribe, and that is a tedious process which might cost users extra

money. Finally, this application does not provide encouragement features to users which might be considered negative.



Figure 2.7 Gooroo application logo.

2.2.2 Our Proposal Versus Previous Work

In this section, we will clarify what distinguishes this work from what we previously found in the relevant work section and highlight some unique areas in which our work draws its strength. This work aims to deliver the most enjoyable and easiest experience for users in which their time, money, and effort are all saved.

Based on what we found in the similar mobile application section, Judan application uniqueness concentrates on choosing talents and hobbies as a main target rather than the static studying subjects, which provides an entertainment experience for users. This focus adds many encouragement features to the app, like providing certified progress improvement. Another critical difference that adds more uniqueness to this application is the implementation of the recommender system. It uses the user's preferences to suggest the most suitable, relevant, close, and professional trainers. The suggestions will be shown as a list to allow trainees to choose the most likely trainer. This is done by using precise algorithms and techniques to give the ultimate relevance. The user has the option of using the manual searching way as well. Moreover, this work focuses on one region as a start, which is Saudi Arabia, and that will become a significant strength for having a high possibility of finding a trainer to meet in person instead of depending only on online classes. It also provides an electronic system specialized for scheduling appointments and choosing the training type (in-person or online) and the platform in the case of online classes. Another important trait is providing the choice for talented people to perform their talent as a service so people can hire them in their special occasions. Finally, Judan application will support trainer's self-marketing trait by providing a social media-like environment so they can post

their accomplishments pictures and videos, certifications documents, and any textual content to get people's interaction and become widely known.

Table 2.1 compares features between the aforementioned applications and our own application:

Features	TutHive App	QuickTutor App	HalaPro App	Gooroo App	Our App
Enable to log in as a trainer and as a trainee.	✓	✓	✓		✓
Contact tutors/trainers via a specific page for arranging an appointment.	✓	✓	✓	✓	✓
Available in Saudi Arabia.	✓		✓	✓	✓
Provide the two options of online and in-person training.	✓	✓	✓	✓	✓
Use the recommended system approach for suggesting trainers.				✓	✓
Provide a Monthly subscription for the trainees.					✓
Provide a personal page for the trainer to post his/her work.	✓				✓
Show the highest rating tutors/trainers in the same city.	✓	✓			✓
Provide the option of hiring a talented person to perform his/her talent as a service.					✓

Table 2.2 Summarize the differences between training management applications.

2.3 Summary

This chapter covers an overview of the recommender system's background, how it started, and it approaches and discovers the related work. Initially, we began with an overview of the recommender system concept and how it became an essential element in our project. Then we went through personalized and non-personalized recommendations. We review recommender system approaches in detail, also we discussed some of the applications related to our work. Finally, we clarified the differences between this work and the previous related work sections and highlighted some unique areas where our work can highlight its advantages. In the next chapter, we will focus on data analysis and modeling.

Chapter 3: System Analysis

This chapter aims to study and analyze the current system and generate its models. It focuses on studying the system in one piece as a whole. The chapter is divided into four sections: analysis of the survey, requirements elicitation, requirements specification, and developmental (or research) methodology.

3.1 Survey Analysis

An operative application needs to take target users' needs into account. The target users for our work can be talented people who are willing to make an income out of their talents by either performing them as a service or teaching them as trainers (online or personal), or can be interested people who are willing to improve their talents with a trustful expert.

Based on a focus-group survey we made and shared among the western region of Saudi Arabia in September 2020, we received several answers which clarified the problems and needs in this area. The number of participants has reached 270. Most of them are young-adults between the age of 20-29 years old as shown in Figure 3.1. The majority of them are Saudis with a massive percentage of 93.3%. Most of the participants are from the western region of Saudi Arabia, 84.1% of them from the city of Madinah, and 9.6% from the city of Jeddah. 93% of participants are females, and 73.7% are unemployed.

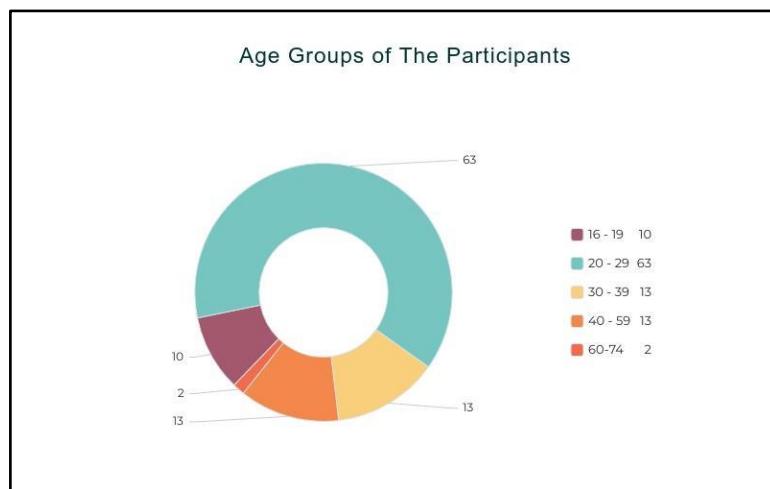


Figure 3.1 Age groups of the participants

Participants were asked to write a hobby they are interested in (not necessarily good at) and a talent they are good at to help us know the potential hobbies/talents categories in Judan application. The top 10 hobbies were: drawing, writing, playing musical instruments, cooking, farming, dancing, digital editing, photographing, sewing, and playing sports games. The top 10 talents were: photographing, singing, cooking, drawing, writing, handmade artworks, digital editing, putting on make-up, diction, and dancing.

We also asked participants to write the difficulties they faced finding a good trainer to improve their talent. The answers summed up for several reasons:

- Some participants face difficulties with training prices
- A couple of trainers demand costly training.
- The talent in which some trainees demand is unique and not familiar, as some said that they want to learn the piano and violin, but there are no trainers specialized for them.
- Part of participants referred to the problem of not having available places for training.
- Some also mentioned fear, anxiety, and lack of confidence in the trainer's training.
- Several participants wrote that the external trainer's timing does not correspond to the local time.

Thus, this confirms that most people have faced issues affecting their talents. The following Figures captures some survey results in which our study will focus on:

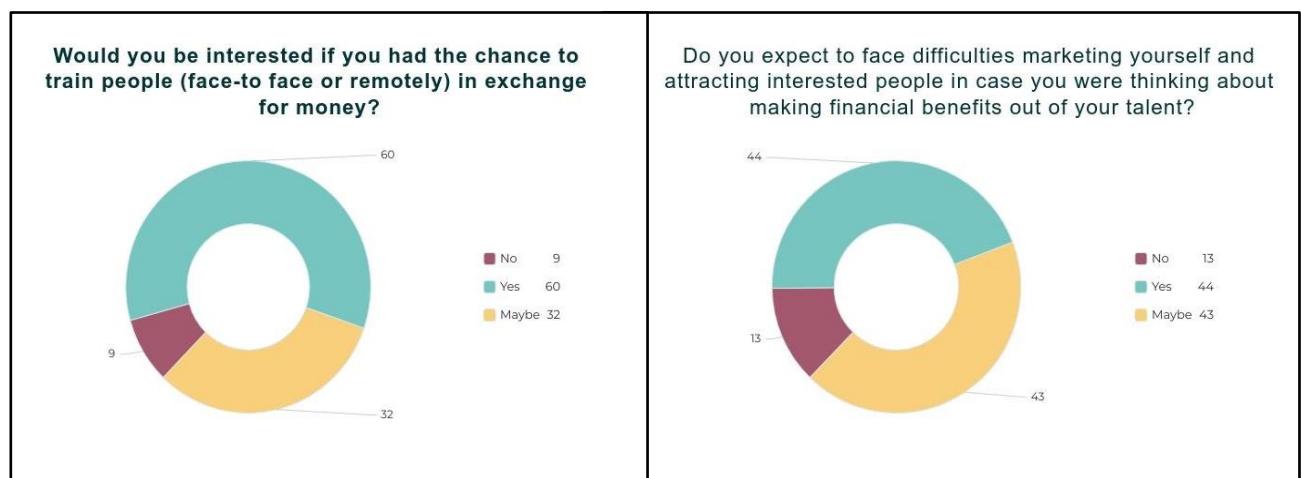


Figure 3.2 Potential trainers' and their expectancy of facing self-marketing difficulties.

As shown in Figure 3.2, we asked participants if they would be interested if they had a chance to train people (personally or remotely) in exchange for money. We got 59.5% with “yes” as an answer, while 31.9% answered “maybe”. The number of people who answered “no” is small-scale compared to the interested people, and that shows the potential users and customers of Judan application. Their interest in training people is considered one of the primary keys of our work success since the first step of starting any successful business is to analyze the demand of the study. We also asked participants if they expected to face difficulties marketing themselves and attracting interested people in case they were thinking about making financial benefits out of their talents. 44% participants answered “yes” and 43% answered “maybe”. This illustrates the need for marketing facilities which allows trainers to share their achievements with their potential trainee to improve the trust and willingness.



Figure 3.3 Participants' recognition of the existence of unqualified trainers.

As shown in Figure 3.3, we asked the participants whether they think that some trainers do not have sufficient experience. Most of them (64%) stated that they think some trainers are not qualified. Approximately 33% of respondents think perhaps there is. However, the remaining 3% disagree with the statement and believe that all trainers have enough knowledge to train. Thus, the majority of answers confirms the importance of having an application that provides qualified trainers, where the trainee is entirely confident, whether through a personal page or people's ratings and comments. We also asked the participants if they think trainers' prices (regardless of their field) are often high. Most participants (65%) agreed that the trainers' costs are expensive. Only 1% do not agree with this sentence. Accordingly, this confirms that trainees need a trustful platform with real and appropriate

prices to forbid trainers from asking for high prices, or maybe using alternative methods for paying money, like the monthly subscription for example instead of paying for each hour. Also, active people always search for discounts and coupons as a reward for their loyalty, and that is not available in most cases.

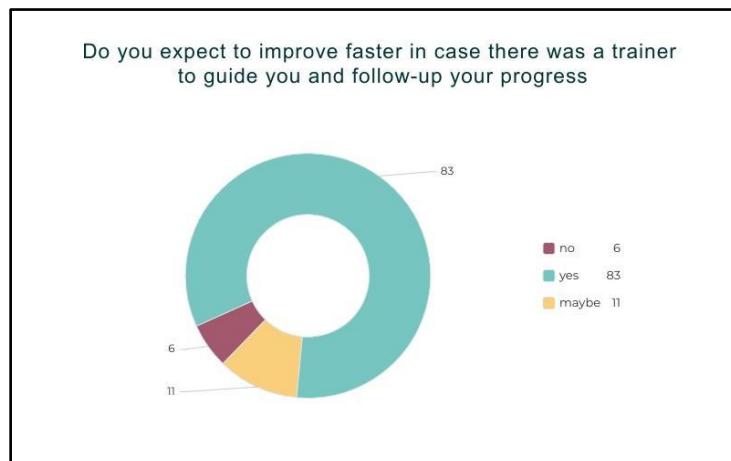


Figure 3.4 The impact of having a personal trainer in the speed of improving a hobby.

As shown in Figure 3.4, the participants were asked whether they expect to improve faster and gain the most of their hobbies in case there was a trainer to guide them and follow-up their progress. 83% strongly agreed that they expected to improve. However, only 6% of respondents disagreed with the suggestion. This underlines the importance of having a personal trainer to achieve their goal, as sticking to a schedule gives them the motivation to follow through. Also, the trainer can set realistic goals based on skills and can adjust whenever it is necessary.

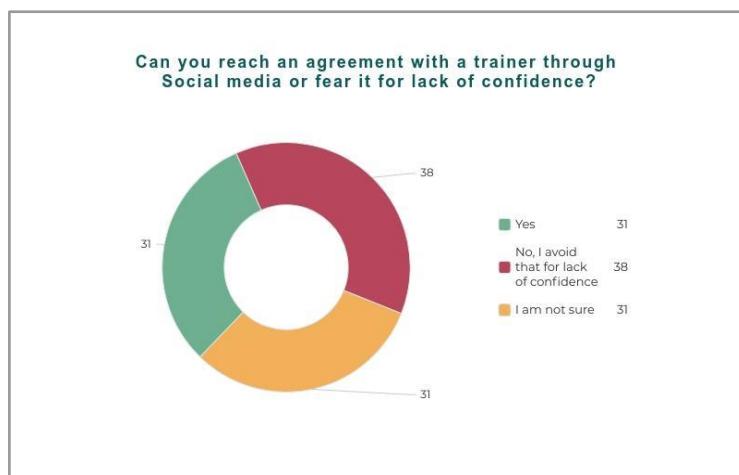


Figure 3.5 The confidence in reaching an agreement through social media.

As shown in Figure 3.5, we asked the participants whether they had previously reached an agreement with a trainer or were suspicious due to a lack of confidence. Most of them 38% answered that they avoided that due to the lack of confidence, while 31% responded that they were not sure (they had never experienced this). Thus, this proves the importance of building a program that facilitates the trainee's access to the trainer and establishes a relationship of trust between them.



Figure 3.6 The difficulty of finding a close trainer.

As shown in Figure 3.6, the number of participants who had difficulties finding a close trainer is 41.1% and it is considered a big percentage, while other participants have never tried to find a trainer. Which leads us to the possibility that there are many people who might be interested in having an application to facilitate the process of matching trainees with nearby trainers.

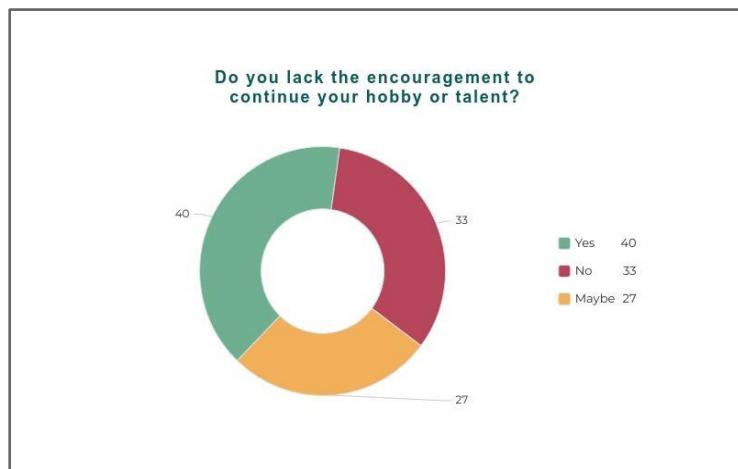


Figure 3.7 Lacking the encouragement needed for the improvement of hobbies.

As shown in Figure 3.7, 40% of the participants answered "yes" when we asked them whether they lack the encouragement to continue practicing their hobby/interest, while 27% answered "maybe". We believe that it is better to encourage and empower people with talent and help them develop their skills so they can find a significant benefit from using the application.

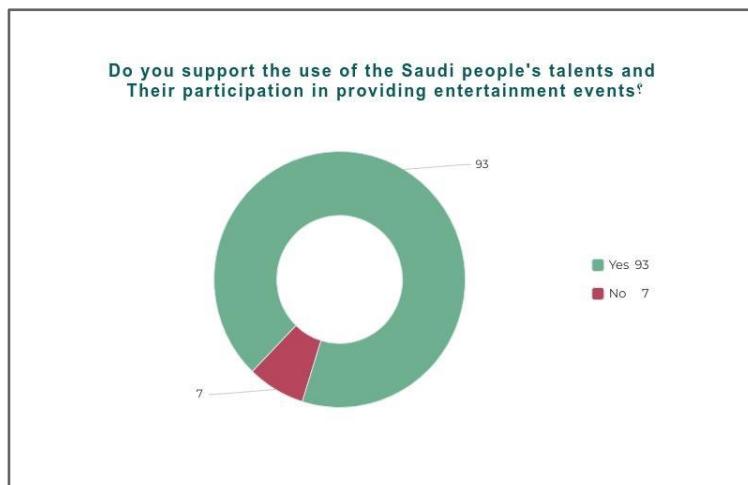


Figure 3.8 The support of Saudi talents.

As shown in Figure 3.8, a huge number of participants 93% agreed on the importance of developing and improving the talents of Saudi people. A key focus for Vision 2030 is creating an environment that unlocks business opportunities, broadens the economic base, and creates jobs for all Saudis. We will achieve this by attracting the best talents and provide the potential for them to make the best out of their talents.

To sum up, the generality supports the exploitation of Saudis talents and their involvement in entertainment activities. Concerning talents, the larger number was interested in teaching talents as trainers (face-to-face or remotely) in exchange for money, and most of them expect to have difficulties marketing for themselves and attracting interested people. Regarding hobbies, the vast majority expect to improve their hobbies faster if a trainer guided them. The top difficulties of finding this qualified trainer were: the inability to find trainers, the high prices of trainers, the disqualification of trainers, the unavailability of time, the transportsations difficulties resulting from the nearby trainers shortage, and the distrust towards trainers. For more details see Appendix A.

3.2 Requirements Elicitation

Requirements elicitation aims to understand stakeholders' work and the functions related to them. This section will identify our system's functional and non-functional requirements based on what we have done in chapters 1 and 2. Also, the result we got from the survey analysis.

3.2.1 Functional Requirement

The functional requirements define the system's functions and describe the system behavior or what the system should do.

1. The system offers two options, either to register (Sign in) using Facebook or Google authentication as a trainer or as a trainee, and then asks the user to enter name, email, password, age, city, country, and interest to create a new user record in the database.
2. The system encrypts user's password using AES decryption method to provide higher security.
3. The system authenticates the user after logging in using a token stored in the device, so the user would not need to login unless the user logged out.
4. If the user already has an account, the system will ask the user to (Login) using Email and password.
5. The system shall only allow users to view the pages stack they need, depending on the type of user (trainee or trainer).
6. The system provides a personal page for each trainer, retrieves trainee's reviews for that trainer, and allows the trainer to manipulate it, and stores the data within each page in the database including shared posts, interests, and profile header information.
7. The system retrieves the registered trainees from the database and displays them to the trainer to view their registration details and upload the certification and update their progress after the session ends.
8. The system allows trainees and trainers to have real-time chat.
9. The system generates trainers recommendations to the specified trainee based on trainers rating and on the matched city and interests. And allows the manual trainers search and retrieve.

10. The system shows the top 6 trainers for each trainee to honor their efforts.
11. The system allows trainees to request the chosen trainer by storing several registration details in the database including date, time, place, training type (online or personal), payment plan, and card details. If the training was online, the place would be (the online platforms like Zoom). If the training was personal. The system will show a map with a marker to be changed by the user depending on the wanted location.
12. The system allows trainees to view trainers' personal page.
13. The system automatically updates trainer's rate in the database when trainee evaluates trainers by calculating the weighted average of the entire rates, and updates reviews stack for trainer when trainee writes a review on the personal page.
14. The system allows the trainee to view the progress and certifications after completing the final training requirements.

3.2.2 Non-Functional Requirements

Non-Functional Requirements defines system constraints and attributes. The non-functional requirements are listed in Table 3.1:

Usability (ease of use)	The system should allow users to perform required tasks easily and efficiently by providing user-friendly UI and flexible UX including the support of both Arabic and English Languages.
Maintainability	The system should consist of modules and should be maintained easily by constantly uploading codes and updating the repository on GitHub using git technology to save changes as commits as well as the serial versions for future enhancement.
Portability	The application can operate on both iOS and Android using mobile cross-platform development framework.
Performance	The system should save trainee time by directly recommending the best match trainer within less than 3 seconds.

Accuracy	Recommendations of trainers should be accurate and relevant to trainee's preferences.
Security	The access authentication is protected using both encrypted passwords and Facebook or Google authentication.
Efficiency	The system should have a fast response time.
Availability	The services should be available and accessible 24/7

Table 3.1 Non-functional requirements.

3.2.3 User Requirements or Domain Requirements

- **User Requirements**

We divided user requirement into two parts: The trainee and the trainer, and we discussed the functions of each type as follows:

1. The trainee

1. The user chooses to sign-up or sign-in as a trainee.
2. The trainee can set his/her personal data (name, email, password, age, city, country and interest).
3. After authenticating the login, the trainee may choose from a set of trainers' recommendations.
4. The trainee chooses a trainer and then requests the chosen trainer by several registration details including date, time, place, and training type (online or personal), payment plan, and card details -optional-.
5. The trainee can evaluate the trainer after the session ends.
6. The trainee can chat with the trainer via private messages.
7. The trainee can view trainers' personal page.
8. The trainee can view the certification after completing the final training requirements.
9. The trainee can view past levels earned from trainers for each subject.

2. The trainer

1. The user chooses to sign-up or sign-in as a trainee.
2. The trainer enters his/her data (name, email, password, age, city, country, and interest). in addition to profile header info (personal photo and bio).
3. Each trainer has a personal page to publish his/her work.
4. The trainer can upload posts and interests on his/her profile.
5. The trainer can view trainees' reviews.
6. The trainer can view the registered trainees' registration details and view their location on map in case it was personal training.
7. The trainer can update the trainee's progress and attaches the certificate after the trainees have completed the final training requirements.
8. The trainer can chat with trainees via private messages.

- **Domain Requirements**

1. Smart hand-held device
2. Internet connection
3. Have Judan application installed
4. Have an account within Judan application.
5. The camera and GPS are allowed to access Judan application when needed.

3.3 Requirements Specification

UML (Unified Modeling Language) diagrams are used to create the models needed in software development. Use case diagram is one of the main UML diagrams, and it is used to show the actors of the system (trainer, trainee, admin), and describe the interaction using a set of use cases as shown in Figures 3.9 - 3.12

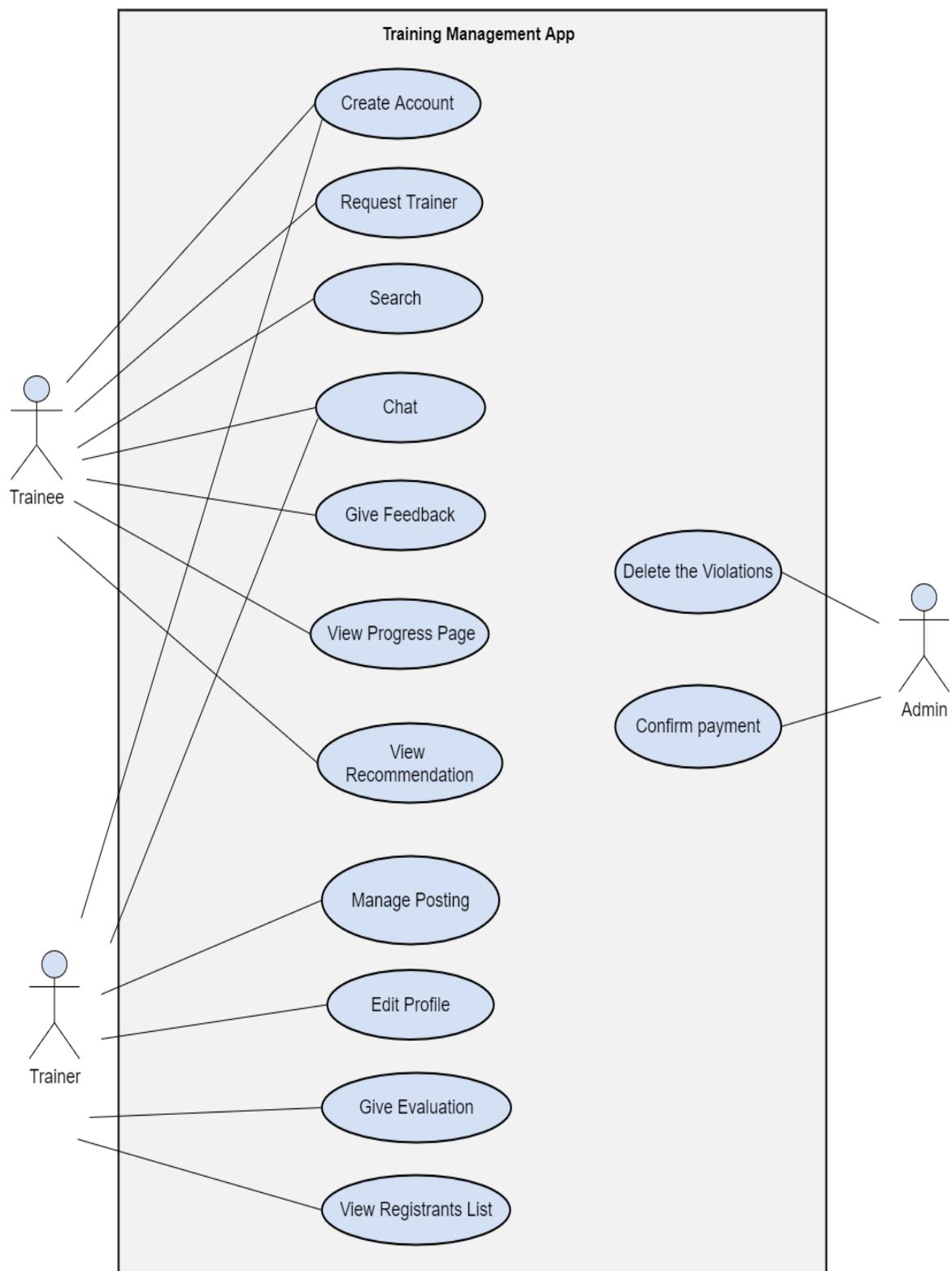


Figure 3.9 Use case diagram

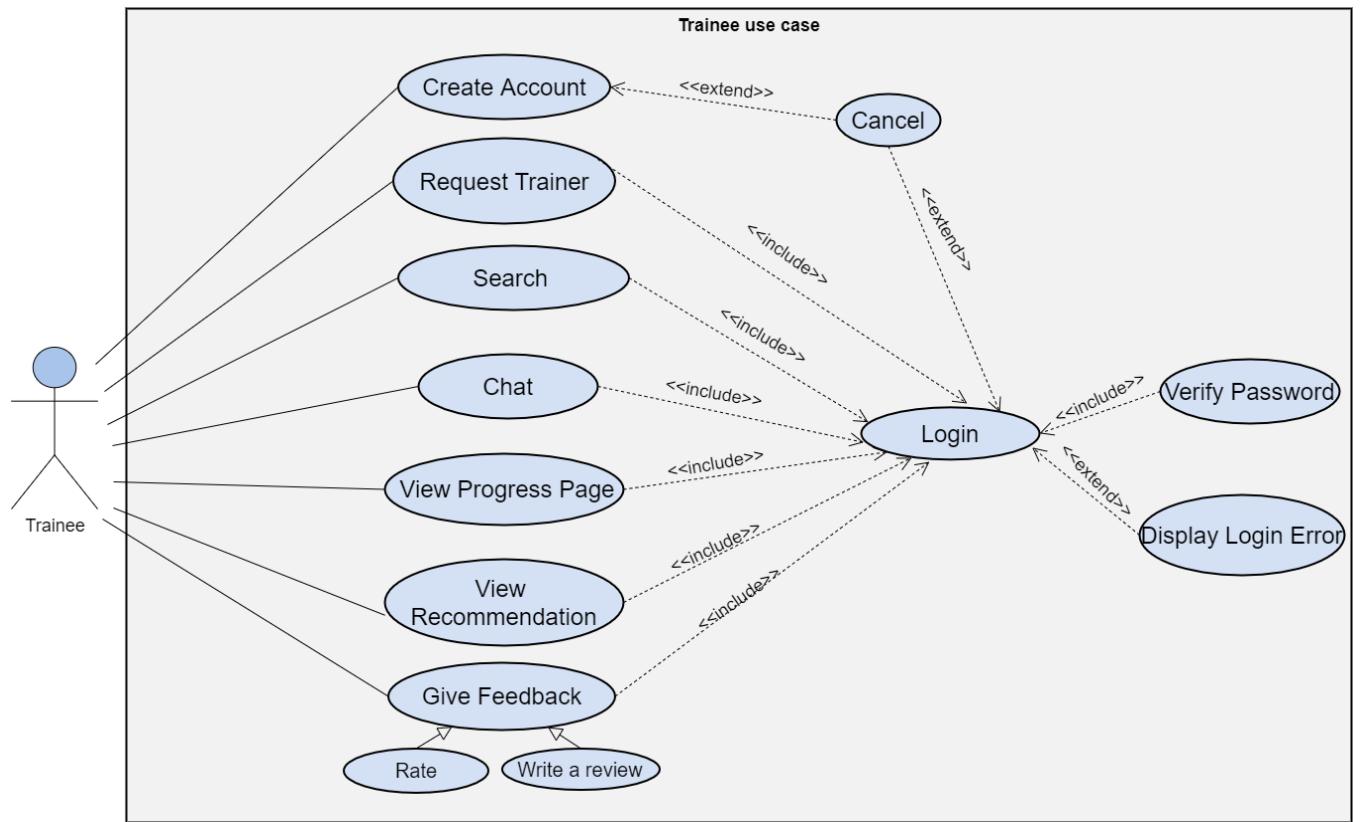


Figure 3.10 Trainee use case.

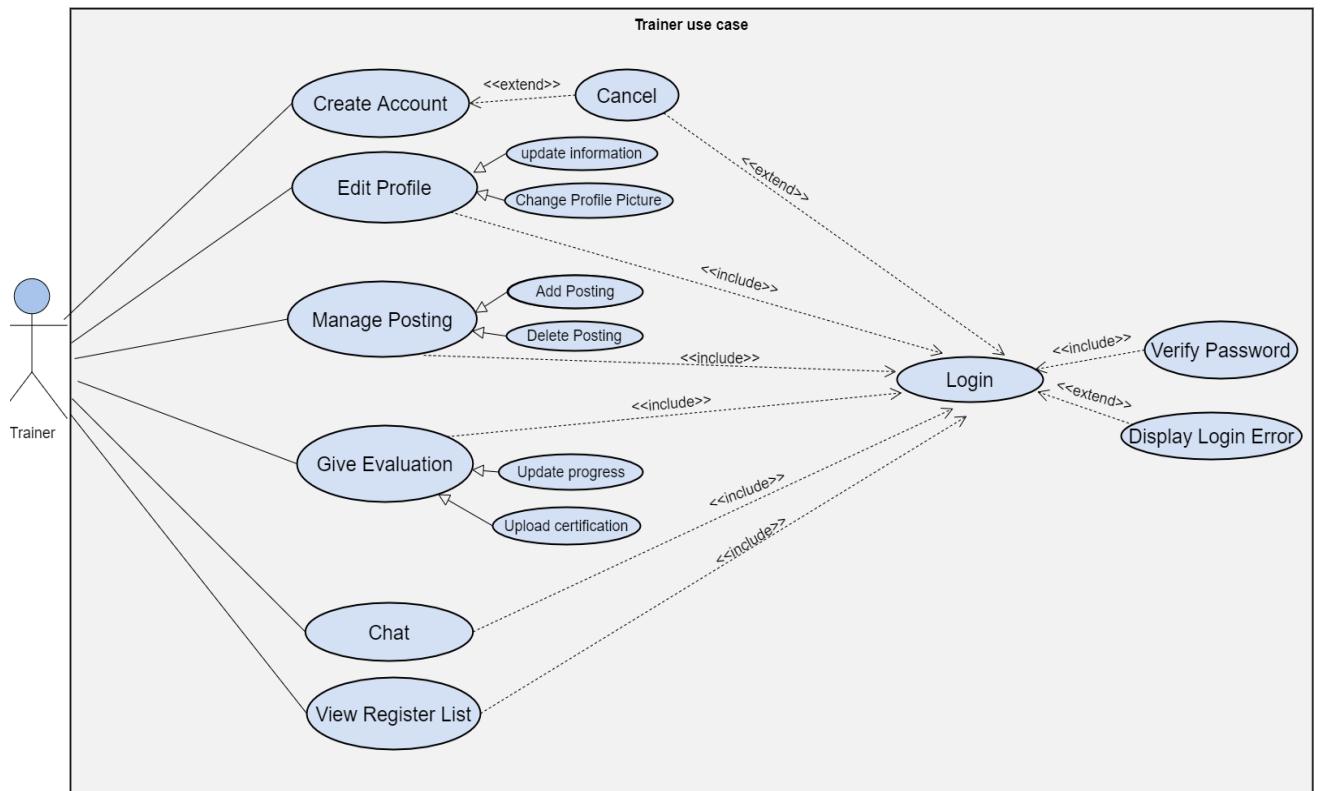


Figure 3.11 Trainer use case

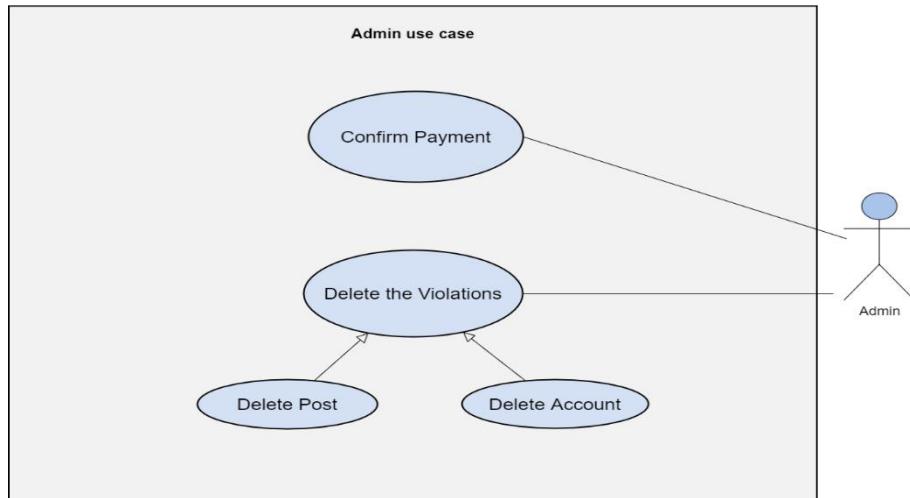


Figure 3.12 Admin use case.

Use Case Name		Create Account
Use Case ID		1
Actors		Trainee/Trainer
Description	Allow users to create an account and become registered users either as trainer or trainee	
Pre-condition	-	
Post-condition	The user account is created with login information, or the account creation request is rejected by the system due to cancellation and errors	
Normal flow	1. User click Create New Account option The list displays the following fields 1.1 Name 1.4 City 1.7 Age 1.2 Email 1.5 Interest 1.3 County 1.6 Password 2. The user enters the required information 3. The system validates the entered values 4. The system encrypts the password 5. The values of user's account information are stored in the database and user's account 6. The system notifies the user that the account has been created	

Table 3.2 Create account use case.

Use Case Name		Search
Use Case ID		2
Actors		Trainee
Description	After creating the account, the trainees can search for trainers using their name	
Pre-condition	The trainee must be logged in	
Post-condition	A list of trainers is presented for trainee to choose from, or an empty page if the trainer does not exist	
Normal flow	<ol style="list-style-type: none"> 1. The trainee must be logged in 2. The system allows the trainee to search for trainers 3. The trainee can send request for the trainer to have a training session, chat with the trainer, or view the personal page 	

Table 3.3 Search use case.

Use Case Name		Request Trainer
Use Case ID		3
Actors		Trainee
Description	When the account is created, the trainees can request a trainer based on their interests	
Pre-condition	The trainee must be logged in	
Post-condition	The trainee must schedule an appointment and choose a training type and place to finalize the requesting process	
Normal flow	<ol style="list-style-type: none"> 1. The trainee must be logged in 2. The system displays a list of recommended trainers to choose from 3. The trainee requests the best match trainer and arrange an appointment 4. The system stores registration details in the database 	

Table 3.4 Request trainer use case.

Use Case Name		Give feedback
Use Case ID		4
Actors		Trainee
Description	After trainees complete the training session, they can rate the trainer and give a review	
Pre-condition	The trainee must be logged in and has the training session completed	
Post-condition	Trainee's evaluation is presented on trainer's page as a weighted average of the entire evaluations (rates)	
Normal flow	<ol style="list-style-type: none"> 1. The trainee must be logged in 2. The trainee has the training session completed 3. The system allows trainee to rate the trainers and post a review about them 4. The system stores the reviews and rates in the database after calculating the weighted average of the rates 	

Table 3.5 Give feedback use case.

Use Case Name		View Progress Page
Use Case ID		5
Actors		Trainee
Description	Trainees can view their progress page to know the current level and view the certification	
Pre-condition	The trainee must be logged in	
Post-condition	The progress level and certifications are viewed, if there is no progress level, the default would be beginner.	
Normal flow	<ol style="list-style-type: none"> 1. The trainee must be logged in 2. The system allows the trainee to view the current level and the certifications owned. 3. The trainee can complete training to level up the progress 	

Table 3.6 View progress page use case.

Use Case Name		View recommendation
Use Case ID		6
Actors		Trainee
Description		Trainees can view recommendations about trainers based on the data gathered
Pre-condition		The trainee must be logged in
Post-condition		The trainee can either choose from the recommendation a trainer or search for a trainer manually
Normal flow		<ol style="list-style-type: none"> 1. The trainee must be logged in 2. The system starts gathering data based on the data gathered 3. The system show recommendations for the trainees as cards 3. The trainee can choose a trainer and send a request

Table 3.7 View recommendations.

Use Case Name		Chat
Use Case ID		7
Actors		Trainer/Trainee
Description		A method of communication between the trainer and the trainee
Pre-condition		Trainee and trainer must both be logged in
Post-condition		The user sees the request on chat list page and starts the chat
Normal flow		<ol style="list-style-type: none"> 1. The trainee and trainer must be logged in 2. The trainee sends a message to the trainer and vice versa. 3. If the trainer or trainee replies and the system keeps tracing of the chat communication in the database

Table 3.8 Chat use case.

Use Case Name	Edit Profile
Use Case ID	8
Actors	Trainer
Description	After the trainers create their account, they can edit their profile, adding their personal photo, field, and biography
Pre-condition	The trainer must be logged in
Post-condition	The new updates are saved and represented on the profile
Normal flow	<ol style="list-style-type: none"> 1. The trainer creates an account and login 2. The system allows the trainers to edit their profile 4. The systems updates these information in the database.

Table 3.9 Edit profile use case.

Use Case Name	Manage Posting
Use Case ID	9
Actors	Trainer
Description	Trainers can manage posting for their personal profile
Pre-condition	The trainer must be logged in
Post-condition	The updates are represented on trainer's profile
Normal flow	<ol style="list-style-type: none"> 1. The trainer must be logged in 2. The system allows the trainers to manage their posts 3. The trainer can add new posts and interests 4. The updates are represented on trainer's profile 5. The system stores these updates

Table 3.10 Manage posting use case.

Chapter 3: System Analysis

Use Case Name	Give Evaluation
Use Case ID	10
Actors	Trainer
Description	After the trainers finish their sessions, the trainers can give evaluation for the trainees on their progress page
Pre-condition	The trainer must be logged in and trainee must has the session completed
Post-condition	Trainer's evaluation is updating in trainee's progress page
Normal flow	<ol style="list-style-type: none"> 1. Trainer must be logged in 2. Trainee finishes the session 3. Trainer can upgrade trainee's level on the progress page and attach certificates to download or share

Table 3.11 Give evaluation use case.

Use Case Name	Delete Violations
Use Case ID	11
Actors	Admin
Description	The admin can manage posting by deleting inappropriate posts, and deleting accounts that violate rules
Pre-condition	To access the system as an admin
Post-condition	Post/Account is deleted, and database is updated
Normal flow	<ol style="list-style-type: none"> 1. After the user registers either as a trainer or trainee 2. The system allows the admin to manage these accounts by deleting inappropriate posts, and deleting accounts that violate rules

Table 3.12 Delete violation use case.

Use Case Name	ConfirmPayment						
Use Case ID	12						
Actors	Admin						
Description	The admin can confirm payment for the trainees						
Pre-condition	To access the database						
Post-condition	The trainees can successfully enroll in the course						
Normal flow	<ol style="list-style-type: none"> 1. After the user register as a trainee 2. The trainee will pick the most suitable trainer for him/her using the recommender system or the manual search. 3. The trainee will fill the request form and choose the time, the training type whether it is online or personal, and the location. 4. The trainee will click on "to payment". 5. The trainee will choose the appropriate training subscription for him/her if the payment is hourly, monthly, or yearly. 6. The trainee will click on the "continue to payment detail" option and fill the fields. The list displays the following fields <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">6.1 Card owner name</td> <td style="width: 50%;">6.4 Date and time</td> </tr> <tr> <td>6.2 Bank name</td> <td>6.5 Upload the receipt</td> </tr> <tr> <td>6.3 Account number</td> <td></td> </tr> </table> 7. The system will display an interface with a message "please wait for receiving a confirmation email for your payment" 8. The admin will confirm the trainee payment from the database. 9. The trainee will successfully be enrolled to trainer's registered list. 	6.1 Card owner name	6.4 Date and time	6.2 Bank name	6.5 Upload the receipt	6.3 Account number	
6.1 Card owner name	6.4 Date and time						
6.2 Bank name	6.5 Upload the receipt						
6.3 Account number							

Table 3.13 Confirm Payment use case.

3.4 Developmental (or Research) Methodology

In this section, we are going to explain our solving methodology in detail by describing the development process model and showing the working procedure used, the process of data collection, the business and technical constraints, and finally the tools and software needed.

The current development approach is the plan-driven approach, more specifically the waterfall model. Plan-driven approach aims to know and predict all the features needed in advance to plan the development phases meticulously. These development phases are planning and analysis, design, implementation, testing, and maintenance. Plan-driven

approach works on the previous software development phases sequentially based on a detailed plan which suits our well-defined project. Our project is divided into chapters with each one focusing on an important development phase. The requirements are approximately fixed and constant and the phases are consistent and predictable because the main idea is unambiguous, and the same team is working on entire phases. Figure 3.13 illustrates the plan-based development phases.

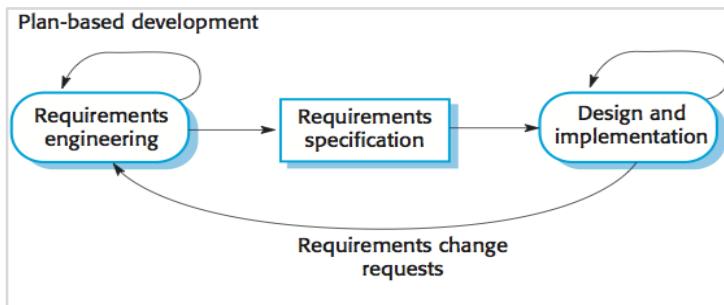


Figure 3.13 The phases of development model [18].

Waterfall model divides the project activities into linear stages. Each stage depends on the earlier stages and should be fully finished before moving to the next one. Figure 3.14 shows the linear development phases in the Waterfall model.

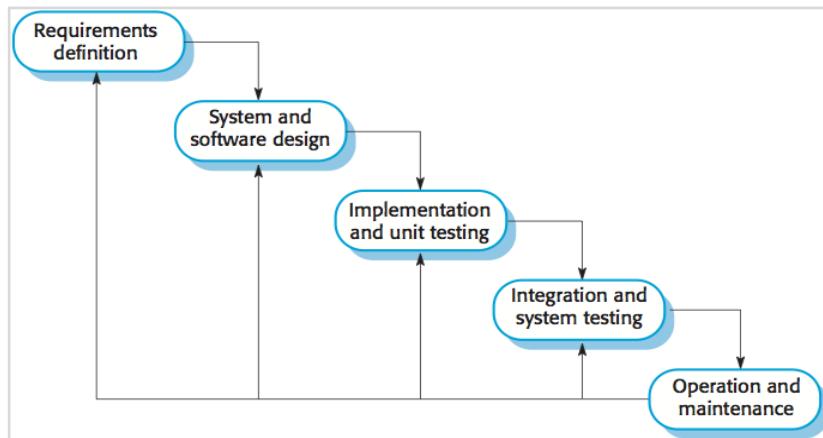


Figure 3.14 The phases of Waterfall model [18].

In our work, the first phase focused on problem definition and data collection from multiple resources (papers, articles, books, surveys, and similar software) to determine the main requirements and needs for Judan application and to know more about which type/s of the recommender system considered the most effective for our use purpose. The second phase illustrated and clarified the functionality of Judan application by representing them using multiple diagrams. The third phase is mainly about designing the system to be user-

friendly and practical. The fourth phase is the beginning of the development phase where the codes are used to produce a tangible software. The last phase is testing the results and periodically maintaining needed updates.

Knowing the limitations and constraints in our project helps us in planning to have the maximum reduction of their effect. Some of the business constraints are firstly the schedule and time, building a fully functional application within less than two semesters is quite challenging; secondly the budget, in which some services needed in development require material charge like the Facebook authentication and having a global hosted server instead of the local host. The technical constraints are mainly the number of available resources, and the experimental or actual dataset availability of trainees and trainers.

Table 3.14 illustrates tools and software used in this project:

Technology	Logo	Use
Google Documents Microsoft Word		Documentation
PowerPoint		Virtual presentation
Google Forms		Survey
Grammarly		Grammar and spelling check
App Store Play Store		Apps search and download
Google Scholar		Scientific search engine
Draw io		UML diagrams
Visme		Charts creation
Adobe Xd		Prototype and UI design
Mysql		Building the database

Android Studio Visual Studio Code Visual Studio		Building apps and code editing
ASP .NET		Back-end API services
React Native		Development framework

Table 3.14 Tools and technologies.

3.5 Summary

In this chapter, we illustrated the functionalities of Judan application in detail, and we determined the functional, non-functional, and domain requirements. In the next section, we presented the UML use case diagrams for the system as a whole and separated for each user, and we described each use case in a table. After that, we demonstrated the development methodology, and we explained the use of the plan-driven approach in our project. We also showed the limitations and constraints of our work and presented the used tools and technologies. In the next chapter, we will talk about the design of the system and how it is organized, and we will show the structure and UI user interfaces of the system.

Chapter 4: System Design

This chapter discusses the structural design of a system, and what the system contains. Firstly, system design is presented using the architectural design. After that, object-oriented design, and its multiple diagrams are discussed. Finally, database modeling design and user interface design is presented.

4.1 Architectural Design

The architectural design is concerned with understanding how a system should be organized and designing the overall structure of that system, i.e., it involves identifying major system components and their communications. Figure 4.1 shows the architectural design of our system.

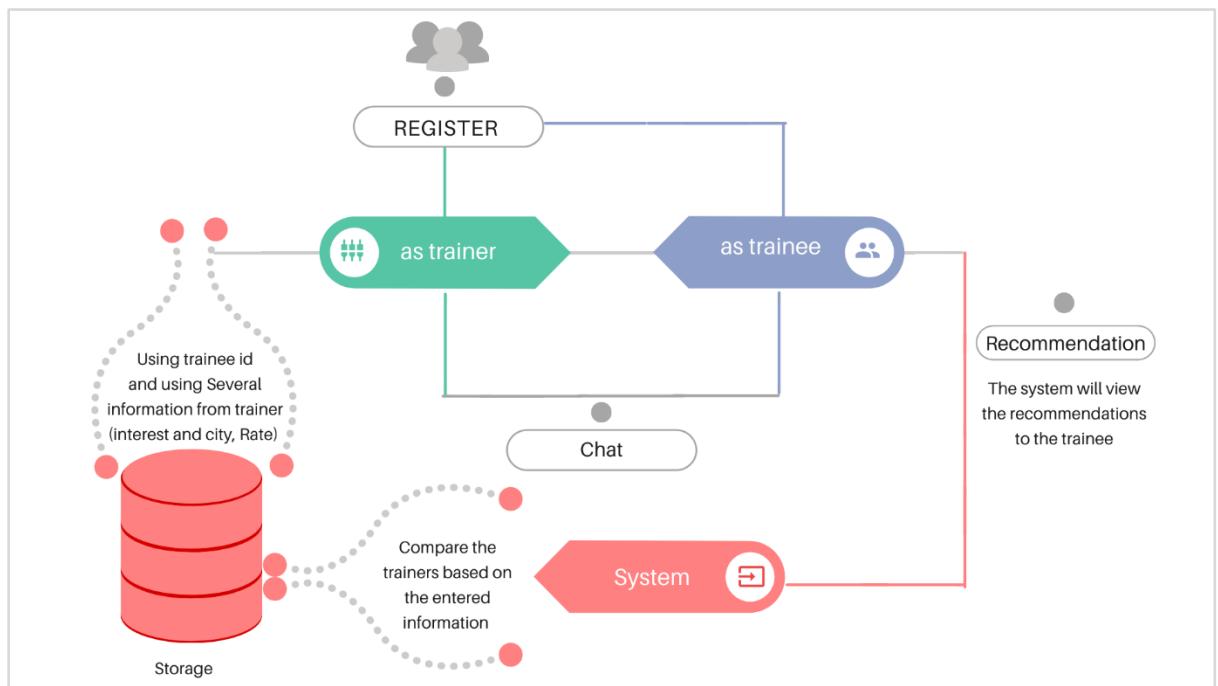


Figure 4.1 Architectural design.

User register either as trainer or trainee, both can perform multiple functions.

The trainee can request a trainer after viewing a list of recommendations driven from the database, or search for a specific trainer manually. The recommendation process depends on trainers' rate arranged in descending order, the shared interest, and preferably the shared location. These information come from the database, and the comparison action is implemented in the API.

4.2 Object Oriented Design

In order to facilitate the planning phase of the software development, we need to show the interrelated objects and clarify the bond between them. Several UML diagrams are used for this purpose including the structural and the dynamic models. The difference underwrites the importance of having both view scopes to have a comprehensive look of the system.

4.2.1 Structural Static Models

- **The Class Diagram**

Class diagram is one of the main UML diagrams, and it is used for describing the structure of the system by showing the basic classes and clarifying attributes and functions within each. This diagram is also used to show relationships between different objects by determining the name of the relationship, multiplicity, and notations. Figure 4.2 shows the main classes of our system. User class is the parent class for trainer and trainee classes, which inherit user class attributes and functions. Trainer is attached with the profile class he/she has. The trainer can also view the registered list of trainees who have chosen that specific trainer. In addition, Trainer has a relationship with the trainee's progress page that allows him to edit the progress and upload the certifications. Trainee class has a relationship with the progress page class that allows viewing the page. Finally, recommendation class takes trainer class objects as an input to process the recommendations and display the customized output for a specific trainee.

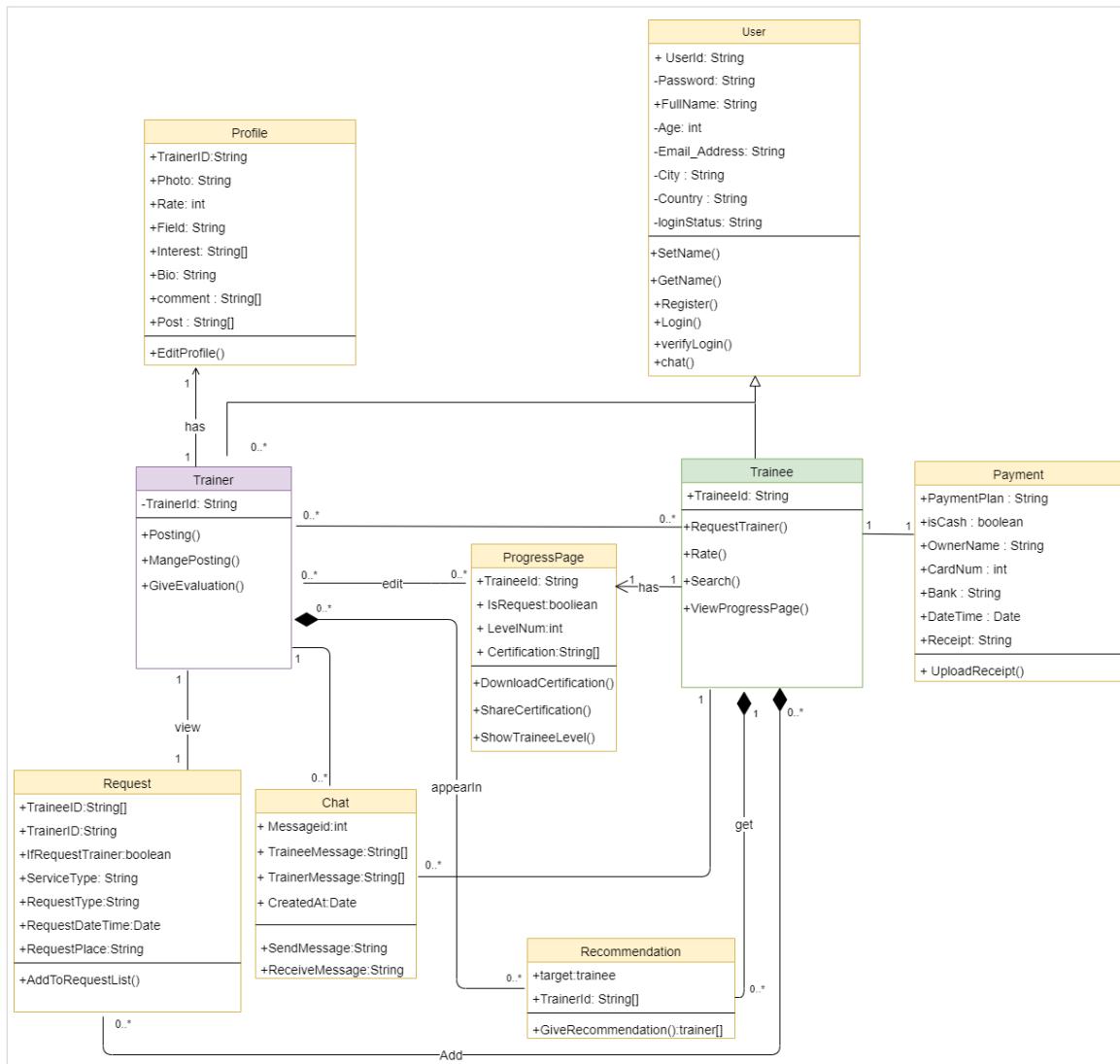


Figure 4.2 The class diagram.

4.2.2 Dynamic Models

Dynamic models used to explain the behavior and functionality of a system over time. They show the functional changes in the system that do not remain in a steady state.

- **The Activity Diagram**

Activity diagram is a dynamic model that is used to show the control flow from the start circle to the end. It focuses on the consecutive functional activity of the system. Figure 4.3 starts with the sign-up activity and shows the order of sequential functions triggered by trainee and trainer. Trainee's activities are mainly finding a trainer, requesting a trainer, evaluation that trainer, and viewing the progress. Trainer's activity can be training a trainee

or self-marketing activities including editing the profile, sharing posts and viewing registrant list.

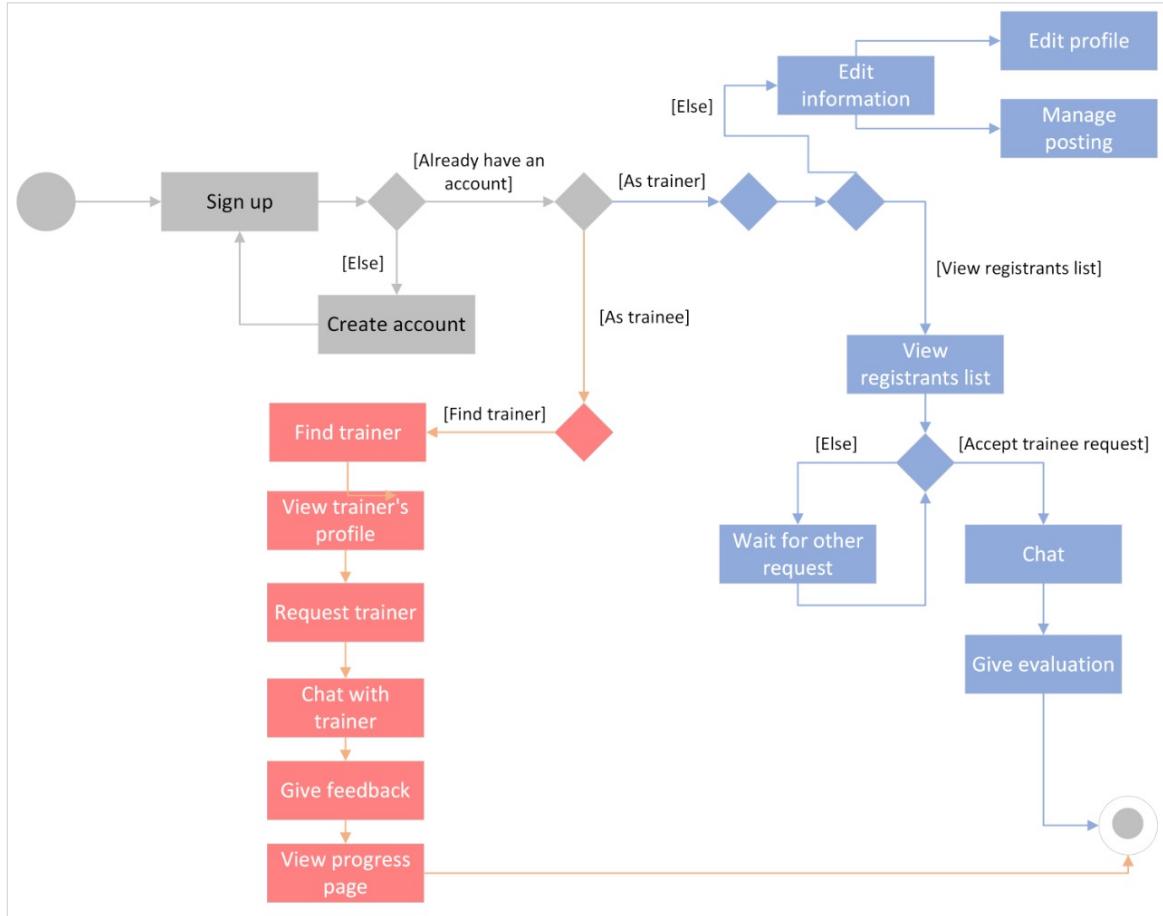


Figure 4.3 Activity diagram.

• The Sequence Diagram

The sequence diagram is an UML diagram that shows the temporal ordered interactions of an operation between objects. It usually takes several objects and specifies the sequential functions of them over time. Figure 4.4 shows the sequence diagram of the trainee actor who wants to get some trainers' recommendations. The trainee begins by signing in Judan application and waiting for database confirmation to show up on the user interface. After verifying the trainee, the trainee will ask the recommender system to match him/her with several ideal trainers. The recommender system will pass trainee's id to the database to return similar trainers. The recommendation system takes trainers' information from the database and implements the recommendation algorithm on them to display the final recommendations to the trainee. These recommendation steps will keep repeating until the trainee chooses the most ideal trainer.

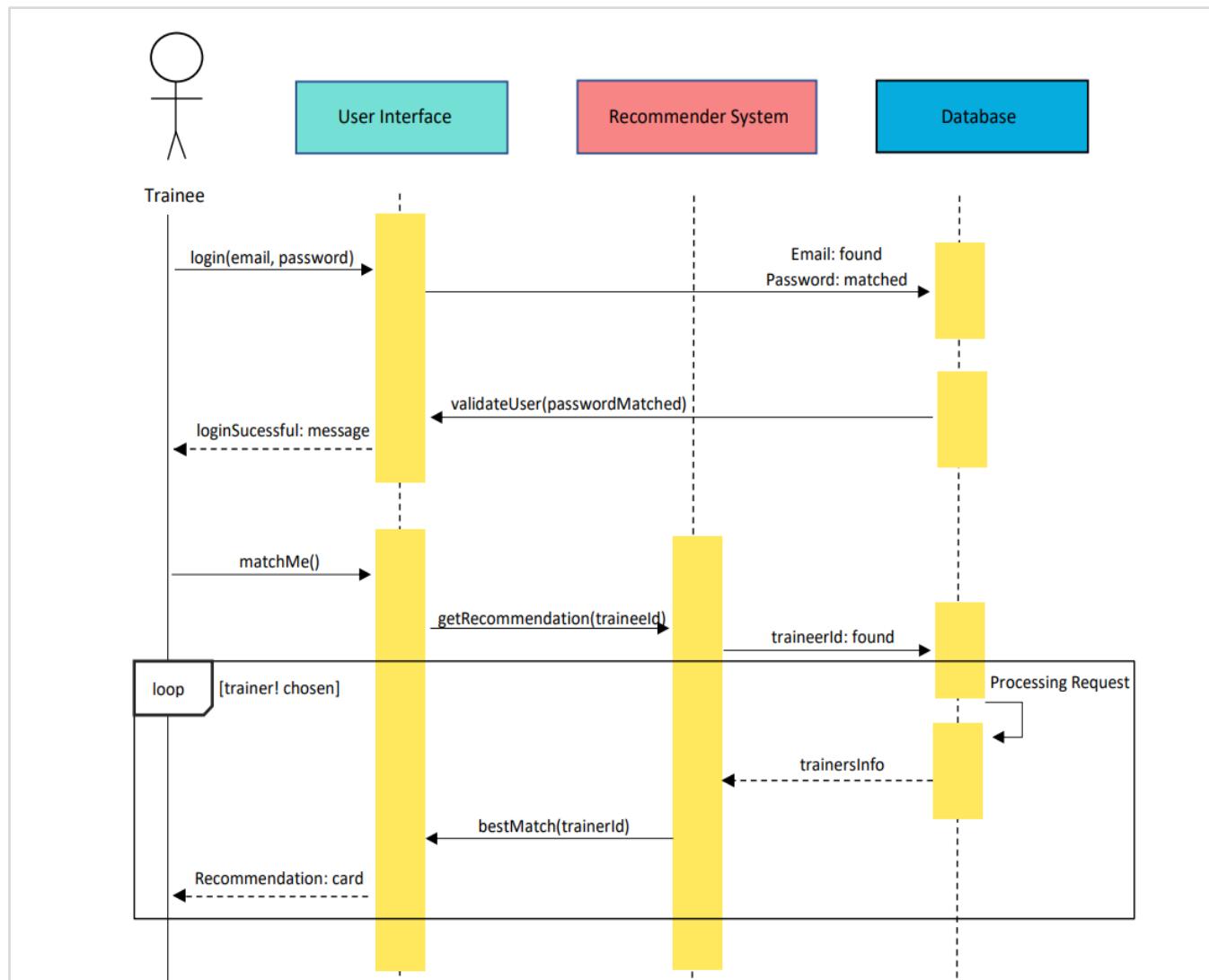


Figure 4.4 Recommender sequence diagram.

Figure 4.5 shows the sequence diagram of the trainer actor who wants to edit the trainee's progress page. The trainer begins after login in Judan application. After this the trainer can click on the progress page interface to edit the trainee's progress page based on their progress improvement (Beginners, intermediate, advanced, expert). The database will save the changes and update the trainee's progress immediately and redirected the trainer to the progress page. The trainer has the ability to upload the trainee's certificate after they are completing the course.

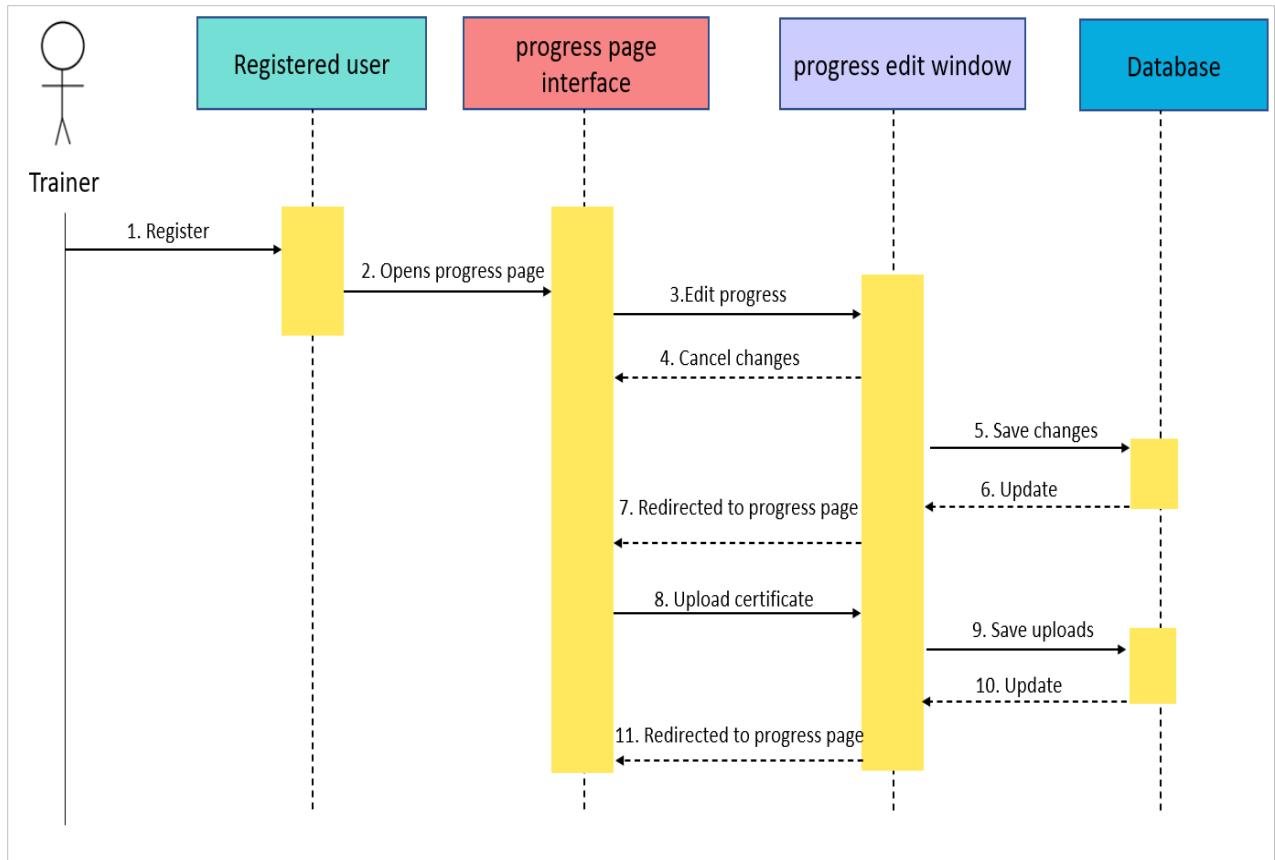


Figure 4.5 The sequence diagram.

4.3 Data Modeling

The database is one of the primary aspects of a complete application. It provides the backend support for accessing related data. Designing the database and clarifying its structure play a major role in every development process.

- **The Database Entity Relationship Diagram**

One of the ways to design a database called entity relationship diagram (ER). ER is a diagram that shows different entities and their attributes, and it distinguishes these attributes by identifying the key type, the composite attribute, and the multiple valued attributes. ER also shows the relationships between entities and their constraints. Figure 4.6 presents the ER diagram of our system.

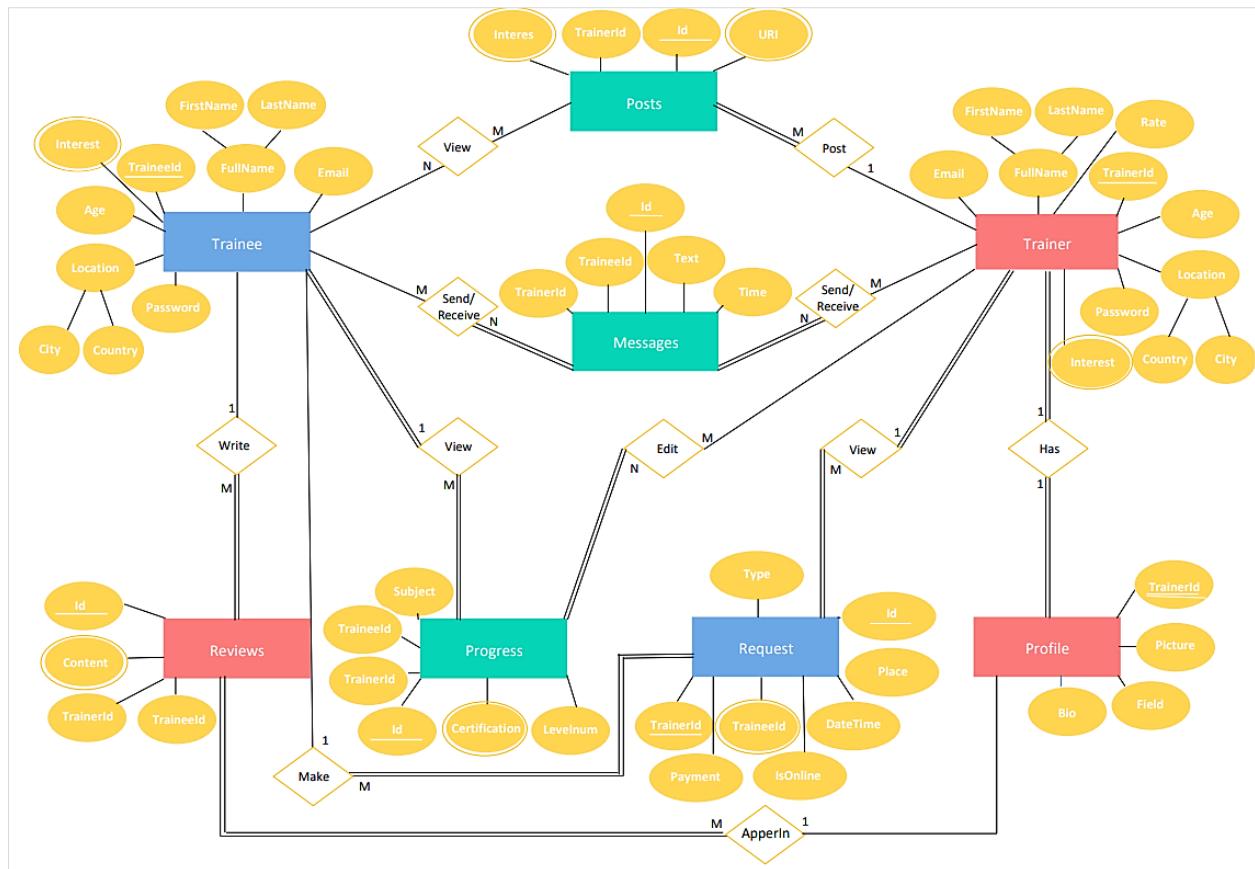
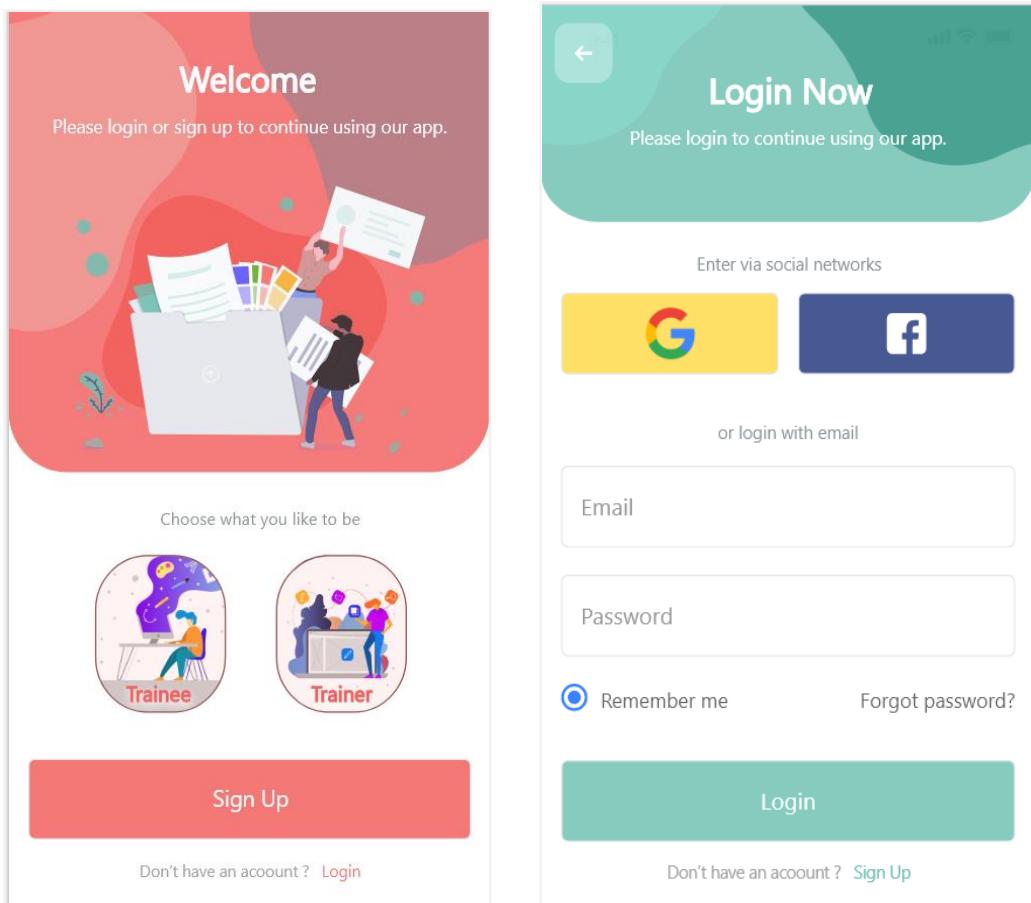


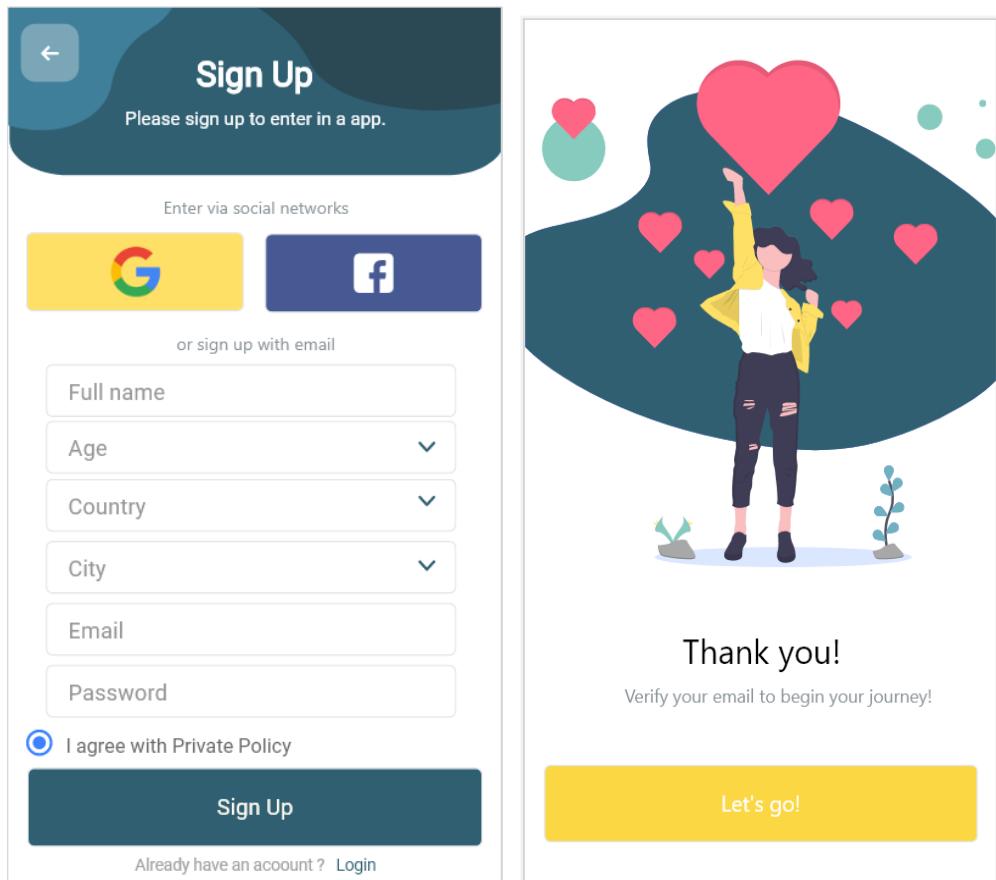
Figure 4.6 The entity relationship diagram.

4.4 User Interface Design

User Interface (UI) is virtual screens that allow users to view or interact with. UI should be easy to access and use and should consider the importance of having a good user experience (UX) to give users the best environment.

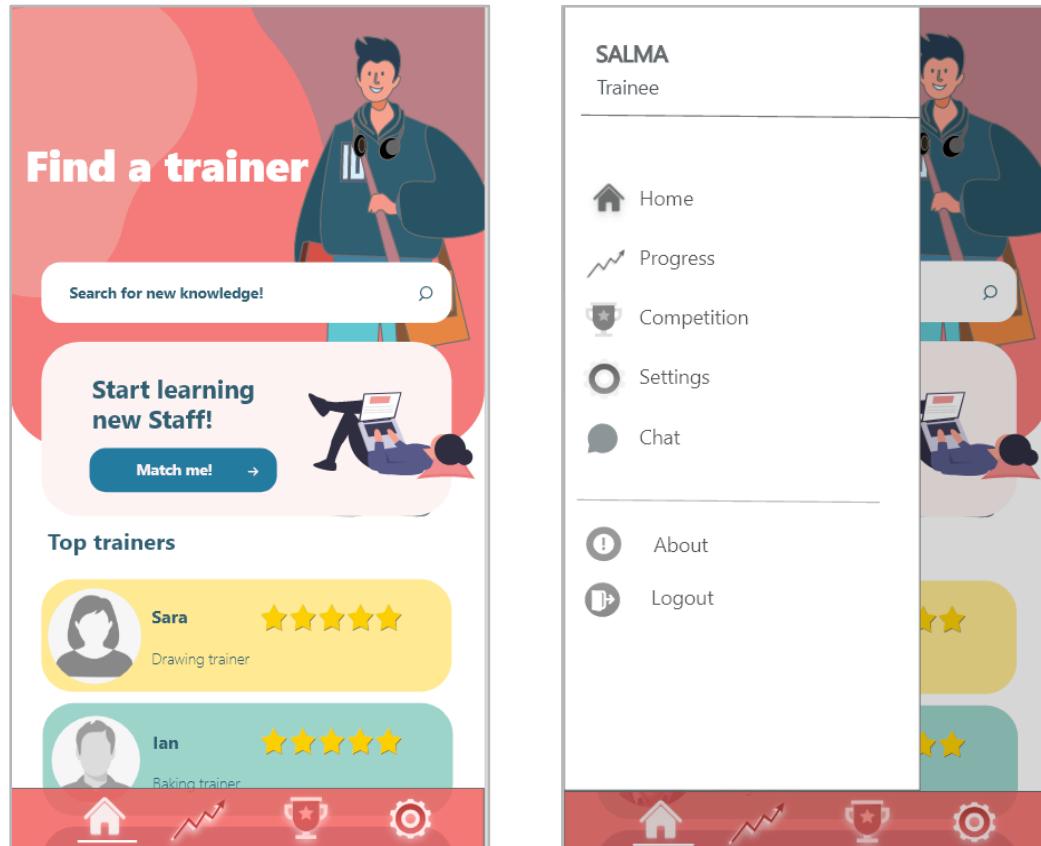
- **Sign up and login**





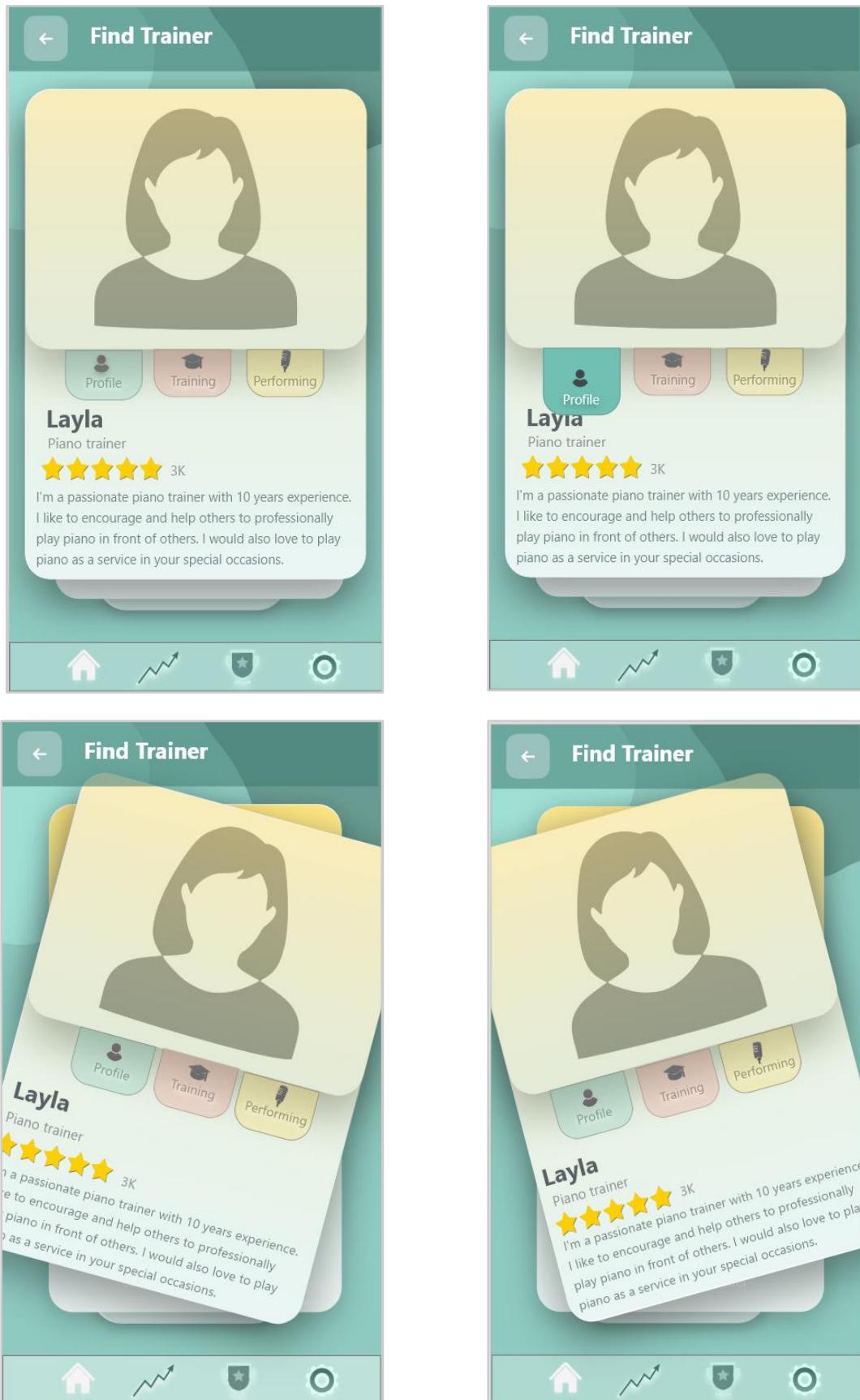
◆ Trainee side

- Home page

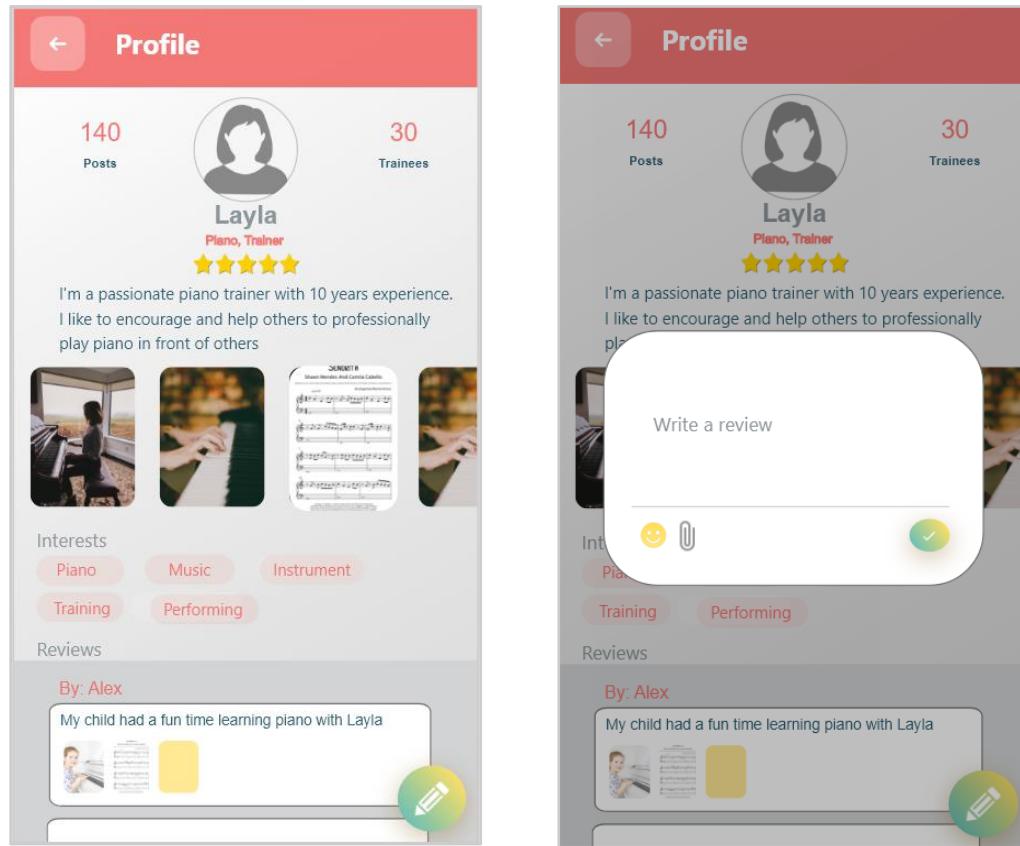


The image displays two screens of the app for the trainee side. The left screen is the 'Home page', featuring a large red banner with the text 'Find a trainer' and a cartoon character. Below it is a search bar 'Search for new knowledge!' and a button 'Match me!'. A section for 'Top trainers' lists two profiles: 'Sara' (Drawing trainer) with a 5-star rating and 'Ian' (Baking trainer) with a 5-star rating. The right screen shows a profile for 'SALMA' (Trainee). The sidebar menu includes 'Home', 'Progress', 'Competition', 'Settings', 'Chat', 'About', and 'Logout'. The main area shows a cartoon character working on a laptop. At the bottom are navigation icons for Home, Progress, Competition, and Settings.

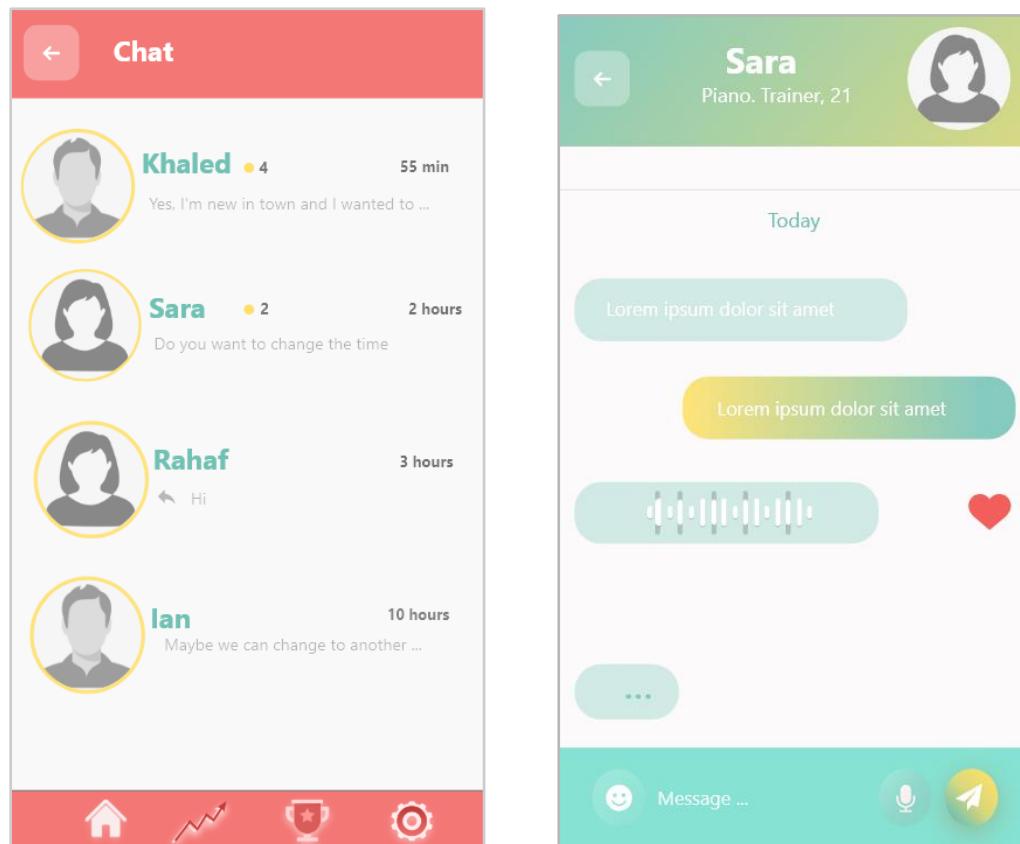
- **Recommend**



- **actions**
- **Trainer profile page**



- **Chat with trainers**



- Request form

Request Form

Type

Online

Personal

Place

Zoom

Date & Time

Date of training

From To

Continue to payment

Request Form

Type

Online

Personal

Place

Date & Time

Date of training

From To

Continue to payment

Pick Your Plan

Hourly Payment
Pay for each hour

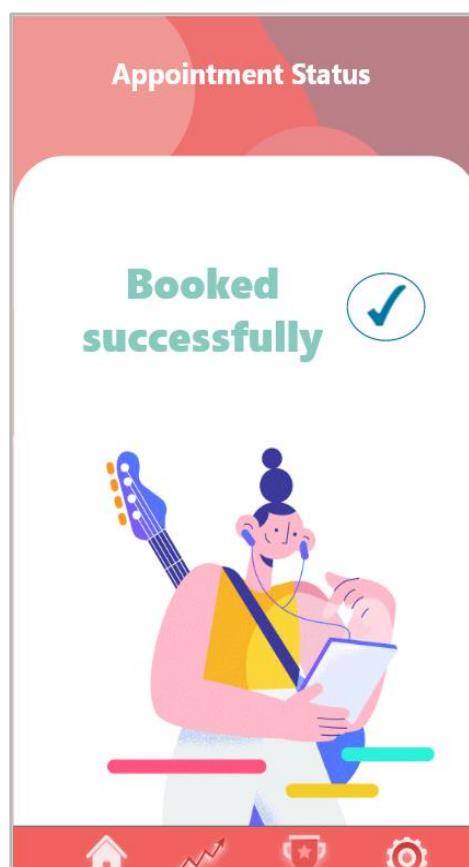
Monthly Payment
5 hours/mo.

Yearly Payment
10 hours/mo.

Trainer card number

Continue with online payment

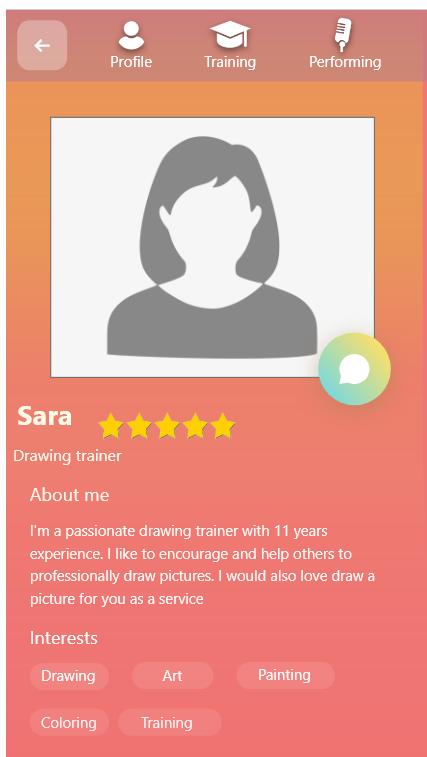
Skip for later cash payment



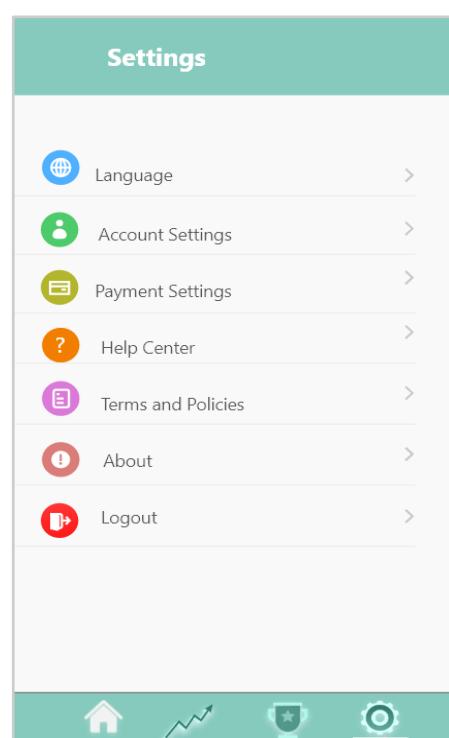
- Progress page



- Trainer Card

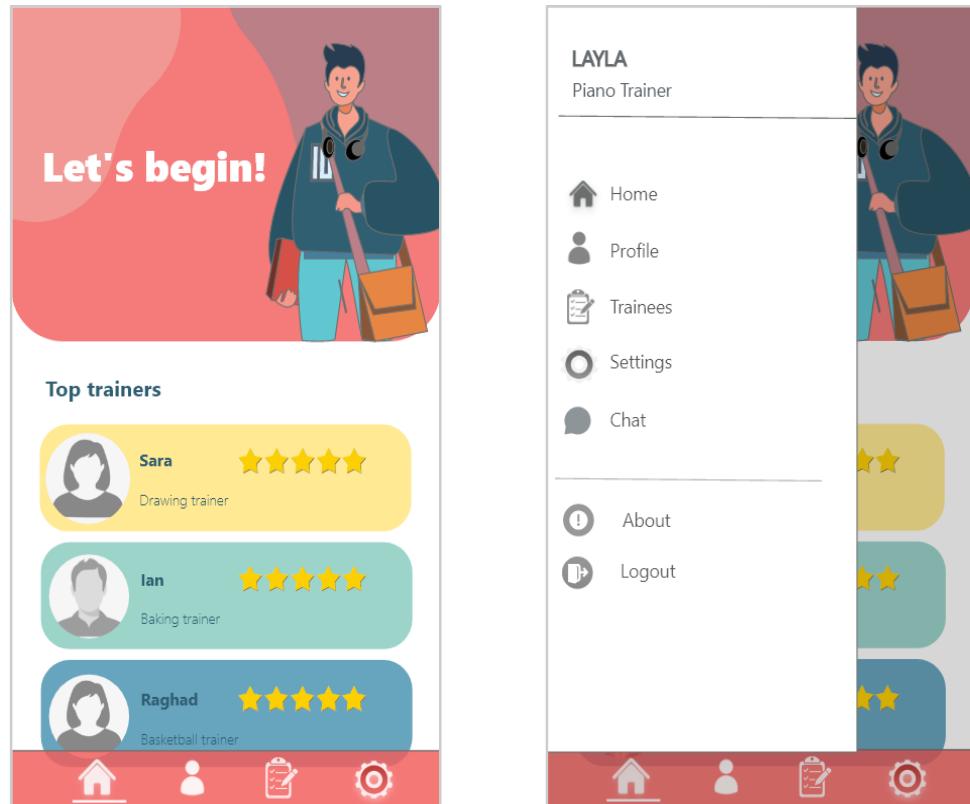


Settings Page

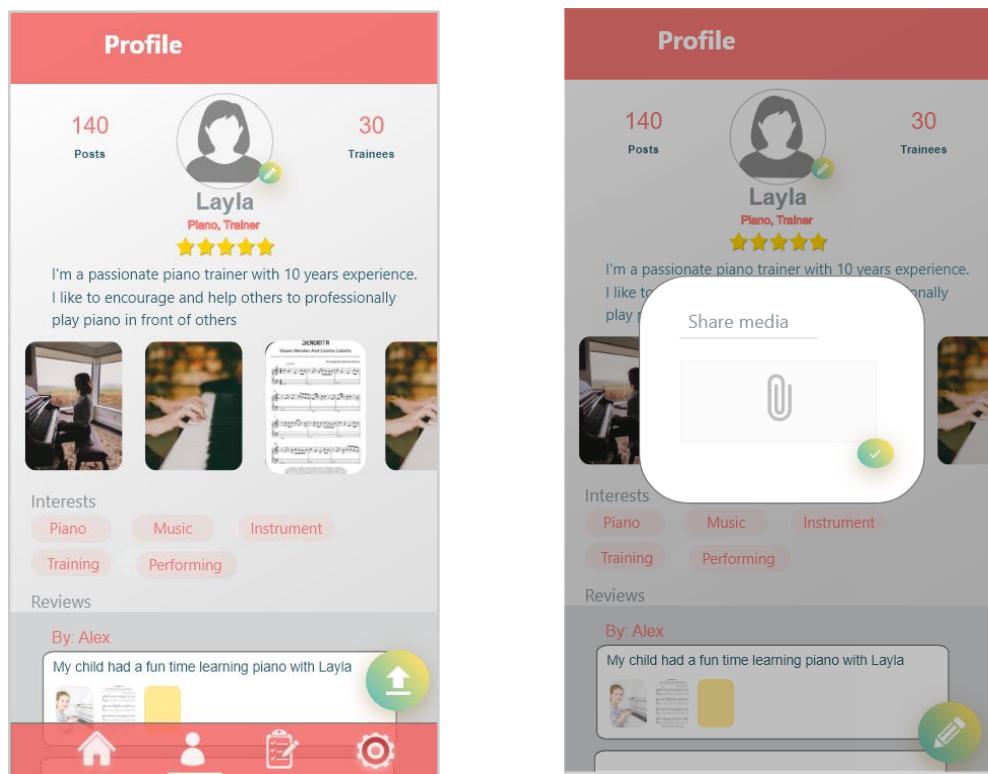


◆ Trainer side

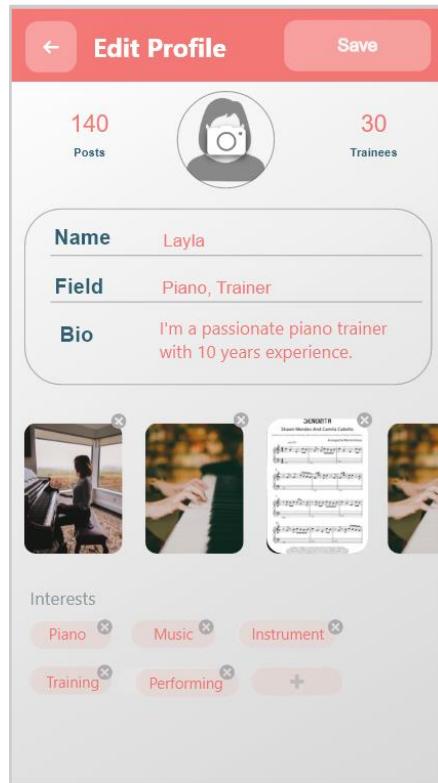
- Home page



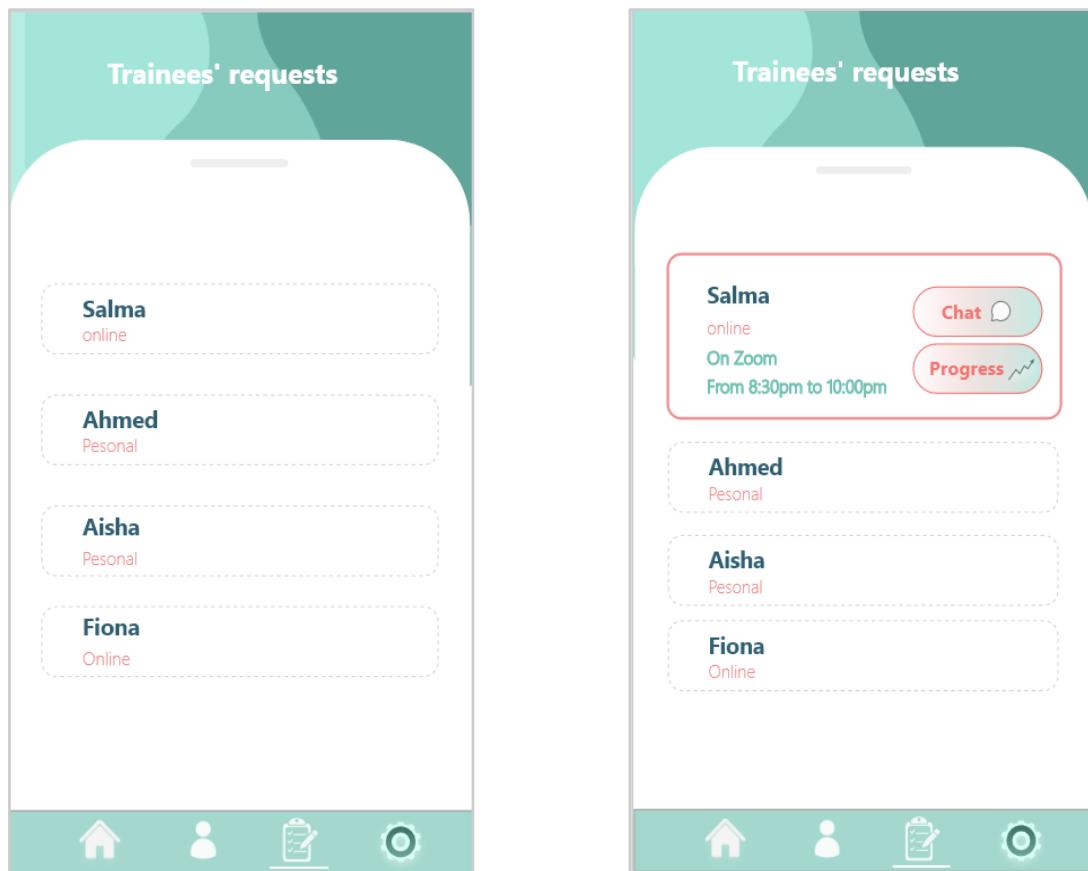
- Profile page



- Edit profile page

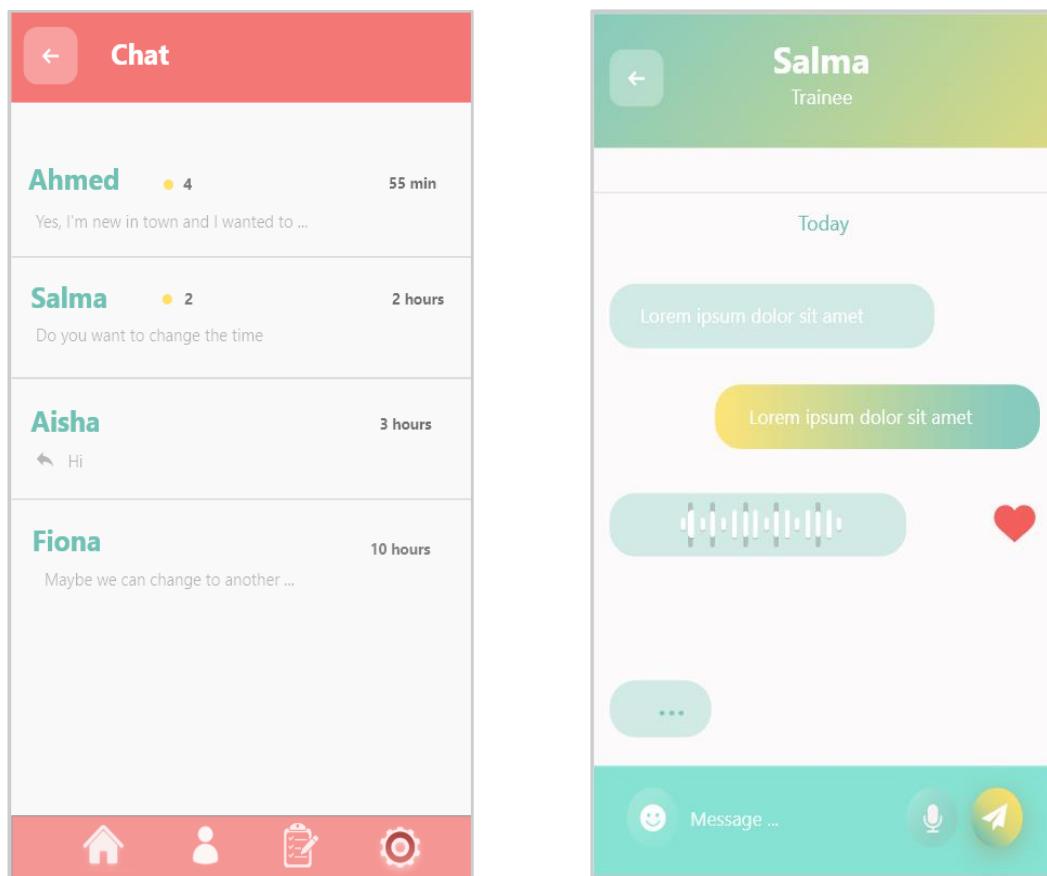


- Trainee's requests page

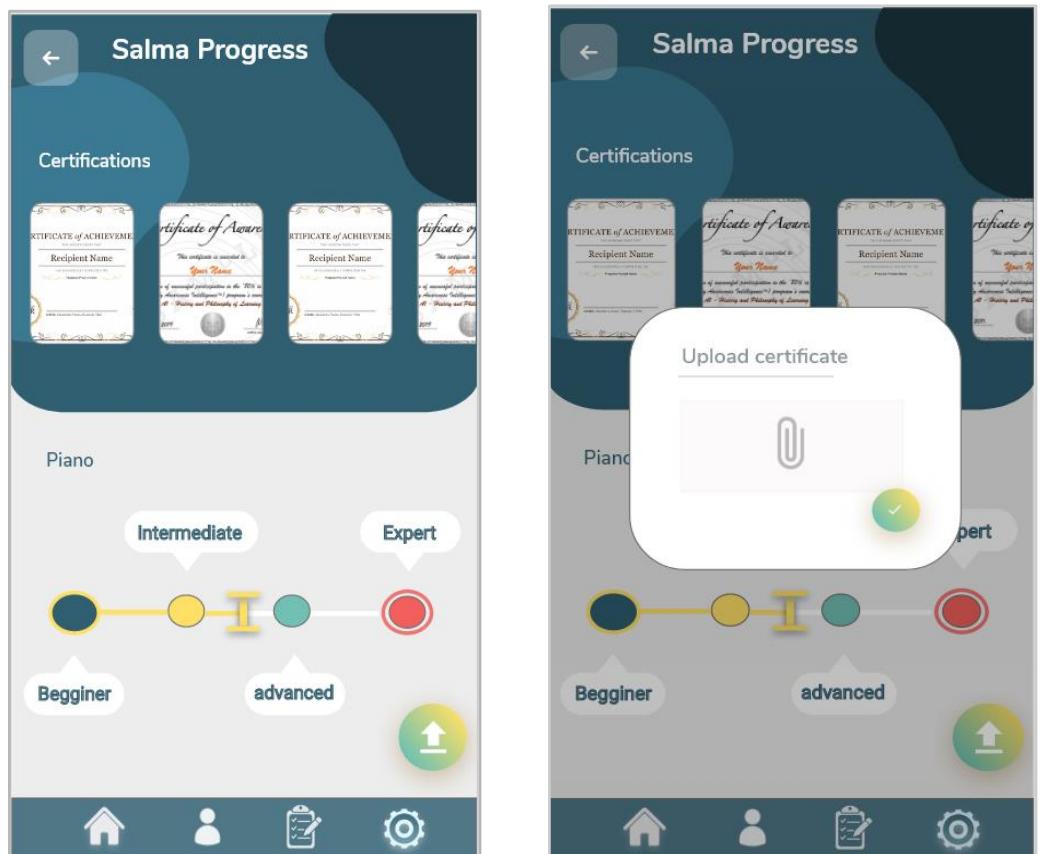


The 'Trainees' requests' screen displays a list of trainees with their names and status. On the left, Salma (online) and Ahmed (Personal) are shown in dashed boxes. On the right, Salma (online, On Zoom, From 8:30pm to 10:00pm) has a red border around her card, indicating an active session. Below her are Ahmed (Personal), Aisha (Personal), and Fiona (Online). Each trainee entry includes a 'Chat' button with a speech bubble icon and a 'Progress' button with a line graph icon. The bottom navigation bar contains icons for Home, Profile, Tasks, and Settings.

- Chat with trainees



- Edit trainee's progress page



4.5 Summary

In this chapter, we presented the overall design of our system including the architecture design, the static models (class diagram), the dynamic models (sequence diagram), the database design (ERD), and finally the user interface design. Using the previous diagrams and designs, our system became clearer and added valuable improvements to the development process. In the next chapter we are going to implement those designs to reach the final chapter and write the conclusion of the whole project and determine the future work.

Chapter 5: System Implementation

5.1 Introduction

This chapter will illustrate the implementation phases starting with choosing the tools and languages and ending with having a complete implementation of Judan mobile application (Front-End and Back-End). Only significant code sections will be highlighted and explained in this chapter after mapping the previous UI design into an implementation. The last section of the chapter focuses on testing phase. It will be separated into 3 main phases. The first phase is unit testing, where each function will be treated as a separate and single entity. Then integration testing, where each component (a group of units) will be tested individually. Finally, system testing, where the system as whole is considered as a complete and unified entity.

5.2 Tools and Languages

In this section, we will briefly describe the tools and software that were used during the system implementation. The tools can be divided into five sections: React Native, Node.js, Android Studio, Visual Studio Code, Structured Query Language (SQL).

5.2.1 React Native.

React Native is known as a JavaScript framework for building native mobile applications for Android and iOS. It uses common knowledge of web technologies (like HTML and CSS, and JSX). It is an open-source platform developed by Facebook in 2015 and used in many popular applications such as Instagram, Facebook, and Skype. In React Native, the code is written only once. In other words, it shares the same codebase for iOS and Android apps, and there is no need to learn native languages. In comparison, Native Mobile requires knowledge of one specific language. For example, the Android app requires knowledge of Java while the iOS app requires knowledge of Swift or Objective-C [13].

The React Native architecture is based on Flux. Facebook developed this Flux to build a client-side app to support unidirectional data flow through the application. Flux

consists of three necessary components: the view, the store, and the dispatcher. The primary purpose of creating Flux is to have a simple architecture and to make it easier to maintain and debug the code [13]. The figure 5.1 shows the illustrate the flow of flux.



Figure 5.1 The illustrate the flow of flux.

There are many steps in the flow of flux. The flow starts when users interact with the view. This interaction can be either by user interaction or through a call from a web API. After that, the view sends an action to the dispatcher. The dispatcher's goal is directing the data flow, ensuring that changes and updates are not cascaded. Also, notify the store when the new change occurs. Based on this notification, the store will update the view. There are many benefits of using flux architecture. These benefits include improvements in data consistency and bugs detection. Another advantage is that it can take the advantage of the separation approach to repeat the same component in another place in the code or other app as well [13].

Since its release, React Native has made it easy for startups to launch mobile applications on the market. The platform is excellent and has many features, making it one of the most promising mobile application development frameworks. Suppose you plan to launch a mobile application for your business? If you choose the traditional way (i.e. native apps), you must hire iOS app developers and Android developers separately. Two different teams mean a lot of work, time, and cost. That is where React Native steals the show. Of course, it is a cross-platform framework, so you can build mobile applications for all platforms simultaneously, since its release, React Native has surprised every technology enthusiast in the mobile application industry. Because this programming language satisfies the purpose of its development, that is, overcomes the limitations of hybrid applications. [20]

Table 5.2 A quick comparison of all the four platforms:

	React Native	Ionic	Xamarin	PhoneGap
Purpose	Learn Once, Write Anywhere	Write Once, Run Anywhere	Write Once, Run Anywhere	Write Once, Run Anywhere
Owner	Facebook	Drifty	Microsoft	Adobe Systems
Language Stack	React.js & JavaScript	HTML, CSS, JavaScript, AngularJs, TypeScript	C#	HTML, CSS, JavaScript
Performance	Native Experience	Slow in comparison to React Native	Performance is stable.	Slower in comparison to others.
GitHub Stars	75K	37K	3K	4K
Testing	Real Mobile Devices or Emulator Required	you can test the code by using any browser	Testing can be done using Mobile Apps	Testing can be done using browsers, mobile devices, or emulators.
Hardware Compatibility	React Native itself is capable	Apache Cordova is used.	SDK Kit is required.	It does not offer much control over hardware; therefore, apps might become slower with frequent render updates.
Reusability	The platform-specific code needs to be changed.	Optimum Reusability of Code.	It allows you to reuse 95% of the code.	You can reuse existing native libraries.

Table 5.1 Comparison of all the four platforms.

Why Choose React Native?

1. Development cost: When you plan to build an application for a startup, money is an important factor. If you want to create a mobile app for iOS and Android but are on a tight budget, then a cross-platform mobile app is the right approach.
2. Upgrade after release: If there is a different upgrade plan after launching the first version of the mobile app, native development is best for your business.
3. Real-time updates: With the widespread use of JavaScript, developers can push updates directly to users' phones without going through the application update cycle. Users like this latest version.
4. Apps with built-in React Native are faster: React Native allows application development using a JavaScript-based declarative programming model, which makes the code more concise and easier to understand.
5. Write once and use the code anywhere: This is the best feature so far. Code written in React Native can be used on almost all mobile platforms. Including iOS, Android, Windows, etc. For more details, see Appendix B.

5.2.2 Node.js

Node.js is an open-source runtime platform. It is commonly used in backend services, which mean it allows the JavaScript program to be executed on the web browser. Node.js is designed as a non-blocking which makes the server highly Scalable and fast.

5.2.3 Application Programming Interface (API).

ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. building great websites and web applications using HTML, CSS, and JavaScript. You can also create Web APIs and use real-time technologies like Web Sockets.

5.2.4 Android Studio.

Android Studio is a platform in which users can create applications for Android tablets and phones and more. In Judan application we used the Android emulator to run the react native code.

5.2.5 Visual Studio Code (VS)

Visual Studio Code is one of the most source editors used among developers. It supports various operating systems such as macOS, Windows, and Linux. It also supports most used languages e.g., JavaScript and python. VS can format and highlight the code to make it easier to read and understand.

5.2.5 Structured Query Language (SQL)

SQL is a standard language used to communicate with the databases. It is designed for retrieving, storing, manipulating, and managing the data inside a relational database management system (RDBMS). In Judo application we used MySQL type to store and retrieve the data.

5.3 Mapping Design to Implementation

In this section of the report, we will focus on the project as a practical document rather than a theoretical framework. In other words, the section describes how the system design in chapter 4 is related to the actual implementation. There exists a direct mapping of most of the designs mentioned in chapter 4, except few changes in the payment interface and in trainer's profile page and bottom tab navigator. These changes have been redesigned in order to enhance the user experience.

- **Payment Interfaces**

The image shows two side-by-side screenshots of a mobile application interface. The left screenshot is titled "BankCard" and contains fields for "Card owner Name" (with placeholder "Enter Card Owner Name"), "Bank name" (with placeholder "Choose a bank"), "Account number" (with placeholder "Last Four digits"), and "Date And Time" (with placeholders "Choose a date" and "Choose a time"). Below these is an "Insert" button and a dashed red box for "Press here to upload the receipt". At the bottom is a "Continue" button. The right screenshot is titled "Payment" and lists three payment options: "Hourly Payment 20 Riyal Pay for each hour", "Monthly Payment 80 Riyal up to 8 hours/mo", and "Yearly Payment 800 Riyal limitless hours/year". Below these are "Continue to payment detail" and "Skip for later cash payment" buttons. Both screenshots have a bottom navigation bar with icons for Home, Progress, Competition, and About.

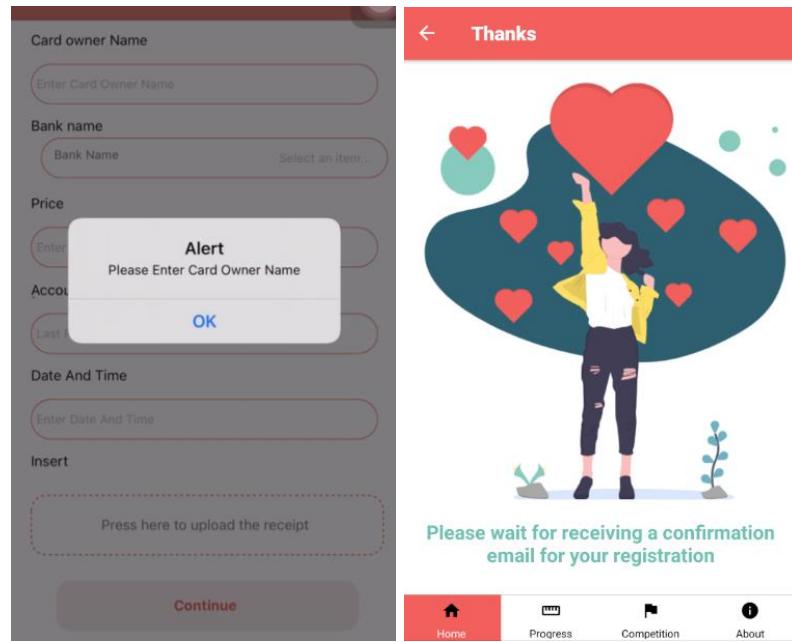


Figure 5.2 Payment interfaces.

In Figure 5.2, the user can choose from multiple plans then continue to payment detail or skip for later cash payment. If the user chooses to continue to payment for more secure and easier way we added a new user interface contain (Card owner name, Bank name, Price, Account number, Data and Time, insert the receipt) to check the money transition manually. We also added a verification where the user cannot move to the next page unless all fields filled correctly.

• Trainer's Interfaces

Figure 5.3 shows trainer's profile page as a home screen instead of the home page in the design. We think that there is no need for the trainer to have an extra home page that will not do any significant work. The profile page also had additional changes, like adding a small header for editing the profile instead the circle that was next to the profile picture to make it clearer and avoid the misunderstanding. We also added the chat tab into the bottom tab navigator to provide easier access.

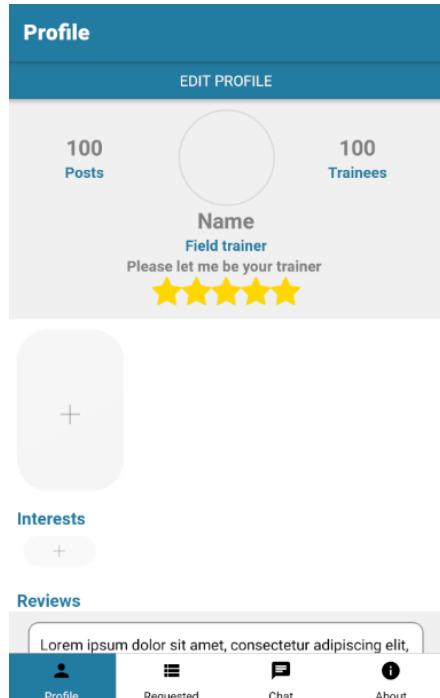


Figure 5.3 Trainer's navigator.

We also added a new sequence diagram, in addition to some changes to the architectural diagram, class diagram, use case diagram, and ER diagram.

5.4 Main/Most Important Codes

In this section, we will go through the most critical functions in the code used for the implementation phase. These functions are recommender system, authentication, home page code and request Page code will be discussed below in more details.

5.4.1 Request Page code

The request form page aims to specify the type, location, and the time of the training. First, the trainee chooses the appropriate type of training for him / her, whether online or personal. If the trainee chooses the type of training to be online, the application will show the available online platforms options automatically. On the other hand, if the trainee chooses to be training personally, the app will show a map of the trainer's location. We have also added a distance constraint. If the trainee's location is more than 80 kilometers away from the trainer's place, the application will reject personal training. Also, the trainee will receive a message the distance is too far. Here is the JavaScript code in figure 5.4 and figure 5.5:

- To show map start point based on trainer's location

```

componentDidMount = async () => {
  const { params } = this.props.navigation.state
  let trainerId = params.id
  let type = params.type
  await this.setState({ trainerId: trainerId });
  await this.setState({ type: type });

  //For map
  let currentCity;
  await axios.get(api + "/idTrainer/" + this.state.trainerId).then(resp => {
    let items = (resp.data);
    items.filter(function (item) {
      currentCity = item.city
    });
    switch (currentCity) {
      case "Jeddah":
        this.setState({ mapData: { latitude: 21.543333, longitude: 39.172779, latitudeDelta: 0.015, longitudeDelta: 0.0121 } });
        this.setState({ initial: { latitude: 21.543333, longitude: 39.172779 } });
        break;
      case "Madinah":
        this.setState({ mapData: { latitude: 24.470901, longitude: 39.612236, latitudeDelta: 0.015, longitudeDelta: 0.0121 } });
        this.setState({ initial: { latitude: 24.470901, longitude: 39.612236 } });
        break;
      case "Makkah":
        this.setState({ mapData: { latitude: 21.422510, longitude: 39.826168, latitudeDelta: 0.015, longitudeDelta: 0.0121 } });
        this.setState({ initial: { latitude: 21.422510, longitude: 39.826168 } });
        break;
      //And the rest of cities ...
      case "Taif":
        this.setState({ mapData: { latitude: 21.437273, longitude: 40.512714, latitudeDelta: 0.015, longitudeDelta: 0.0121 } });
        break;
    }
  })
}

```

Figure 5.4 map start point based on trainer's location.

- To convert latitude and longitude to kilometer

```

//To change red marker position and move the map with it
handleRegionChange = mapData => {
  this.setState({
    markerData: { latitude: mapData.latitude, longitude: mapData.longitude },
    mapData,
  });
  this.distance(this.state.initial.latitude, this.state.initial.longitude, this.state.mapData.latitude, this.state.mapData.longitude, "K");
}

//To calculate the distance from longitude and latitude
distance(lat1, lon1, lat2, lon2, unit) {
  if ((lat1 == lat2) && (lon1 == lon2)) {
    return 0;
  }
  else {
    var lat1 = Math.PI * lat1 / 180;
    var lat2 = Math.PI * lat2 / 180;
    var theta = lon1 - lon2;
    var radtheta = Math.PI * theta / 180;
    var distance = Math.sin(lat1) * Math.sin(lat2) + Math.cos(lat1) * Math.cos(lat2) * Math.cos(radtheta);
    if (distance > 1) {
      distance = 1;
    }
    distance = Math.acos(distance);
    distance = distance * 180 / Math.PI;
    distance = distance * 60 * 1.1515;
    if (unit == "K") { distance = distance * 1.609344 }
    if (unit == "N") { distance = distance * 0.8684 }
    if (distance > 80) {
      alert(AorE.A == true ? LangAr.Distance : LangEn.Distance)
    }
  }
}

```

Figure 5.5 convert latitude and longitude to kilometer.

5.4.2 The weighted average rating

To update the rate for a trainer after each trainee gives a specified rate, we need to calculate the weighted average rates, the maximum result should be 5 and the minimum

result is 0. This result is tightly attached to the number of trainees who rated the same trainer before. The function will take the rating number from the pressed stars component with the user is as a second parameter and use them in the mathematic formula. Total rating variable is the sum of all the previous rates, and overall is trainer's current rate which is fetched from the database. Here is the code see figure 5.6:

```
//To get the current trainee's rate and display it
componentDidMount = async () => {
  let currentRate;
  await axios.get(api + "/IdTrainer/" + this.props.rateForUser).then(resp => {
    let items = (resp.data);
    items.filter(function (item) {
      currentRate = item.rate
    });
    this.setState({ starCount: currentRate })
  })
}

async onStarRatingPress(rating, user) {
  //To calculate the weighted average rate
  let overAll = 0
  let totalRating = 0;
  //To get trainer's current rating
  await axios.get(api + "/IdTrainer/" + user).then(resp => {
    let items = (resp.data);
    items.filter(function (item) {
      overAll = item.rate
      totalRating++
      totalRating += rating;
      overAll = ((overAll * totalRating) + rating) / (totalRating + 1)
    });
    this.setState({
      overAllCount: overAll
    });
    //To show the individual rate
    this.setState({
      starCount: rating
    });
  })
  //To update the rate for the trainer
  await axios.post(api + "/TrainerRateUpdate/" + user, {
    rate: this.state.overAllCount
  });
}
```

Figure 5.6 The weighted average rating.

5.4.3 Recommender System Code

Content-based recommendation systems recommend an item (trainer) to a user (trainee) based upon the similarity between item and user information. Several information (interest and city) are entered by the user, and (Rate) is commonly learned from the feedback that users provides on items. A variety of matching algorithms have been adapted, and the choice of learning algorithm depends upon the representation of content.

To implement the previous algorithm easily, we divided the implementation into front end and back-end sides. In The back end, we firstly built an API action that can retrieve trainee's information from the database given the trainee id. Furthermore, we build another action that can retrieve trainers who has the same given city and interest, which we retrieved from the first action. These trainers will then be ordered by descending based on the highest rate. The final action is the same as the previous one but only retrieve the highest rated trainers with the same given interest regardless of the matched city. The purpose of having the last function is to use it in the front end to concatenate the trainers array driven from the second action with trainers' array driven from the last action. The final result would be all the trainers who have the same city and interest first ordered descending by the highest rate, then the trainers who only share the same interest but not the city. The city is not regarded in the last action because the user might prefer the online option and the city will not matter.

- **Algorithm**

```

Algorithm: Recommender System
Input: TRAINEE_ID
Output: Array of matched Trainers Trainers[n] {n | n ∈ W}

```

Begin

Initialization

```

Let traineeFeatures = {}
traineeFeatures = Fetch(TraineeGetApi/ TRAINEE_ID)
Let city = traineeFeatures.city
Let interest = traineeFeatures.interest
Let trainers[] = Fetch(TrainerGetApi/ interest)

```

Trainers descending order by rate

```

forall trainer i in trainers[i].do
    If (trainers[i].rate < trainers[i+1].rate)
        Swap trainers[i] and trainers[i+1]
end

```

Compare trainer's city with trainee city

```

forall trainer i in trainers[i] do
    If (trainers[i].city == city)
        Move trainers[i] to the beginning of trainers[]
end

```

end

Return **trainers[]**

Figure 5.7 Recommender System Algorithm.

- The Front End

```

componentDidMount = async () => {
  let userId = await AsyncStorage.getItem("userId");
  let city = ""
  let FirstData = []
  let FinalData = []

  //Get trainee city and interest by using the id
  await axios.get(api + "/IdTrainee/" + userId).then(resp => {
    let items = (resp.data);
    items.filter(function (item) {
      city = item.city;
      interest = item.interest;
    });
    this.setState({ traineeCity: city });
    this.setState({ traineeInterest: interest });
  })

  //Get trainers array who have the same city and interest
  await axios.get(api + "/Match/" + this.state.traineeCity + "/" + this.state.traineeInterest)
    .then(resp => {
      FirstData = resp.data
    })

  //Get trainers array who have only the same interest and add it to the end othe previose array
  await axios.get(api + "/Interest/" + this.state.traineeInterest + "/" + this.state.traineeCity)
    .then(resp => {
      FinalData = FirstData.concat(resp.data);
    })

  this.setState({ data: FinalData })
}

```

Figure 5.8 Recommender System front end.

- The Back End (Server side)

```

[HttpGet("Match/{city}/{interest}")]
0 references
public IEnumerable<TrainerDto> GetMatch(string city, string interest)
{
  return datacontext.Trainers.OrderByDescending(x => x.Rate).Where(x => x.City == city && x.Field == interest).Select(x => new TrainerDto
  {
    Id = x.Id,
    Name = x.Name,
    Email = x.Email,
    City = x.City,
    Country = x.Country,
    Age = x.Age,
    Field = x.Field,
    Rate = x.Rate,
    Bio = x.Bio,
    Picture = x.Picture,
    TraineeNum = x.TraineeNum,
    PostNum = x.PostNum
  }).ToArray();
}

[HttpGet("Interest/{interest}/{city}")]
0 references
public IEnumerable<TrainerDto> GetByInterest(string interest, string city)
{
  return datacontext.Trainers.OrderByDescending(x => x.Rate).Where(x => x.Field == interest && x.City != city).Select(x => new TrainerDto
  {
    Id = x.Id,
    Name = x.Name,
    Email = x.Email,
    City = x.City,
    Country = x.Country,
    Age = x.Age,
    Field = x.Field,
    Rate = x.Rate,
    Bio = x.Bio,
    Picture = x.Picture,
    TraineeNum = x.TraineeNum,
    PostNum = x.PostNum
  }).ToArray();
}

```

Figure 5.9 Recommender System back end.

5.4.4 Authentication

To enhance the security of the Judan application, we apply two type of authentication: Social authentication and encryption method. For social authentication, the user will sign in with social provides, specifically using Facebook login. To do that there are a few steps. To authenticate the users, the application will receive a User token access. The Facebook SDKs will automatically refresh the token whenever the user uses our app. This token will expire after 60 days after last use. After the Authentication, the app will turn the user to sign up page to complete the registration. The code below represents the Facebook authentication.

```
export const facebookLogin = () => async dispatch => {
  let token = await AsyncStorage.getItem("fb_token");
  if (token) {
    dispatch({ type: FACEBOOK_LOGIN_SUCCESS, payload: token })
  } else {
    doFacebookLogin(dispatch);
  }
};

const doFacebookLogin = async dispatch => [
  await Facebook.initializeAsync({
    appId: "1763096950517235",
  });
  const {
    type,
    token,
  } = await Facebook.logInWithReadPermissionsAsync({
    permissions: ["public_profile"]
  });
  if (type === "cancel") {
    return dispatch({ type: FACEBOOK_LOGIN_FAIL });
  }

  await AsyncStorage.setItem("fb_token", token);
  dispatch({ type: FACEBOOK_LOGIN_SUCCESS, payload: token });
];
```

Figure 5.10 Facebook Authentication

On the other hands, we also used encryption method by encrypt user's password on the database using AES algorithm. AES is a symmetric-key algorithm which uses the same key (128, 192 or 256 bits) to encrypt and decrypt (128 bits) block, and then uses the substitution-permutation to each block for 10, 12 or 14 rounds (depending on key size). This key is stored safely, and it is hard to detect. The code below represents the encrypt code that is used in our application.

```

let user = await AsyncStorage.getItem("user");
if (user == "trainer") {
    await axios.get(api + "/EmailTrainer/" + this.state.email).then(resp => {
        let items = (resp.data);
        let pass = this.state.password
        let go = false
        items.filter(async function (item) {
            //decrypting the password
            let bytes = CryptoJS.AES.decrypt(item.password, key);
            let originalText = bytes.toString(CryptoJS.enc.Utf8);
            if (originalText == pass) {
                go = true
                await AsyncStorage.setItem("fb_token", token);
                await AsyncStorage.setItem("userId", item.id.toString())
            }
            if (originalText != pass)
                alert("The email or password not correct");
        });
        if (items.length == 0)
            alert("The email or password not correct");
        if (go == true)
            this.props.navigation.navigate("Thanks")
    })
}
    
```

Figure 5.11 Encryption Authentication.

5.5 System Testing

System testing aims to test each component of the application to ensure that it works properly and completely. In Judan application, the test that was used is a Unit test. The unit testing tests the smallest piece of the code individually to make sure that each software unit operates as planned. Here, we tested all parts of our application for the trainer and the trainee side. The following table will provide the result that we got after testing each unit.

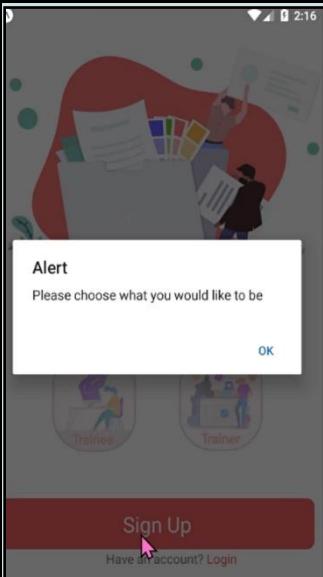
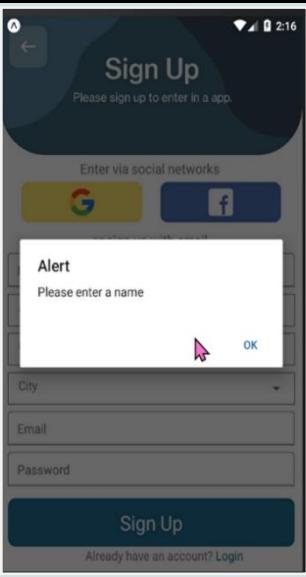
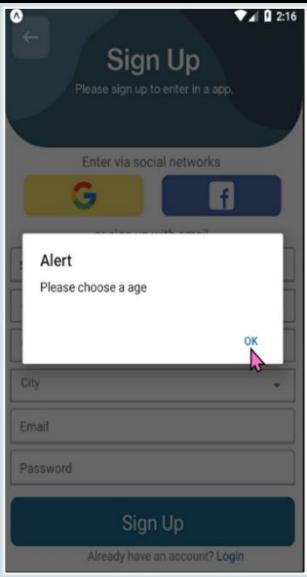
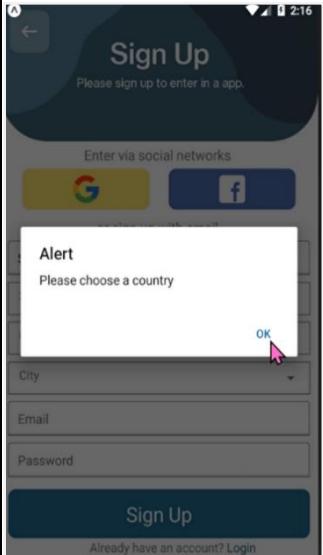
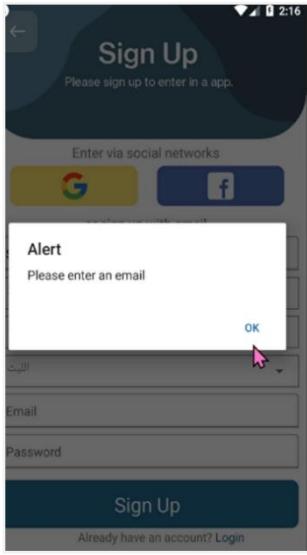
Test ID	Task	Expected behavior	Pass/fail
T1	The user has to choose between the two options, either to register as a trainer or as a trainee before signing up.	The system displays an alert with the message "please choose what you would like to be".	Pass
T2	When a user wants to sign up, the application will check first that all fields filled correctly.	The system will check each field sequentially starting from the first failed and notify the user about the missing or the error.	Pass
T3	When the user entered the password, the application will check that the password at least 7 characters.	The system will check that the password contained 7 characters. If the user entered less than 7 characters, the system would notify the user with the message "The	Pass

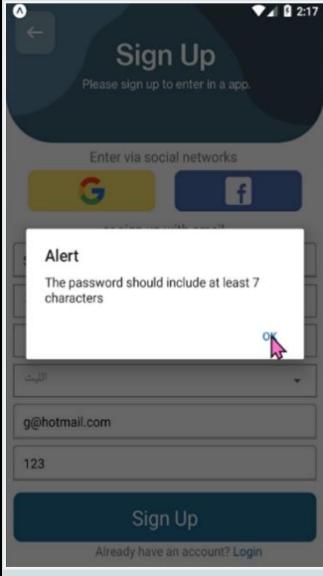
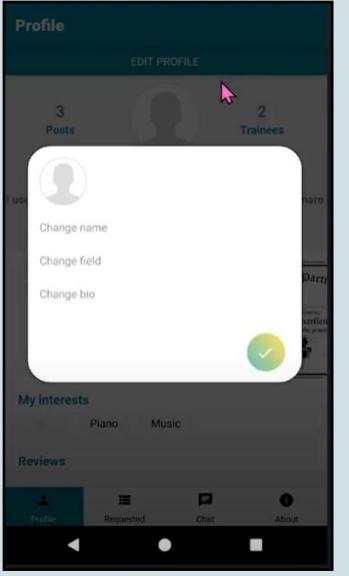
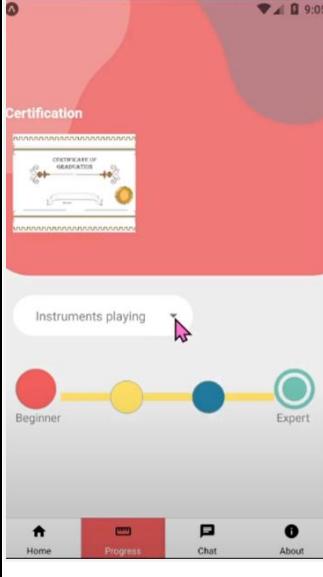
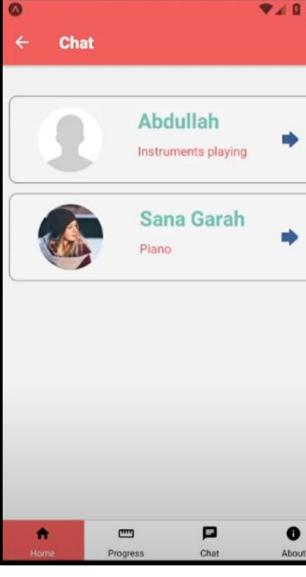
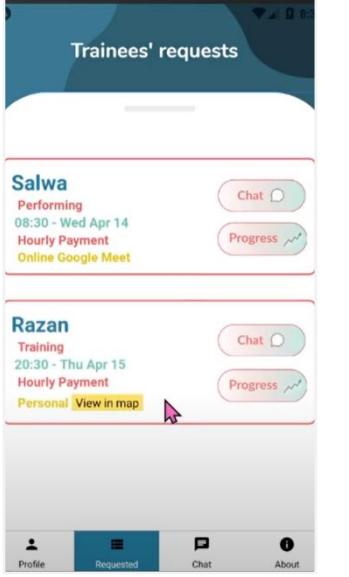
		password should include at least 7 characters".	
T4	The system authenticates the user using Facebook authentication.	The system will turn the user to the Facebook application to verify his/her account. After that the system will turn the user to sign up page to complete the registration.	Pass
T5	Allow the trainer to edit his/her personal page and add posts.	The system able the trainer to edit each his/her profile filed (profile image, bio, interests, and the posts)	Pass
T6	Allow the trainers to evaluate the trainees and upload the certification and update their progress.	The system retrieves the registered trainees from the database and displays it to the trainer to upload the certification and update their progress.	Pass
T7	Allow the trainees and the trainers to have real-time chat.	The system provides a chat icon in the navigation bar to allow the users to click on it and start chatting.	Pass
T8	Allow the trainee to view the progress and certifications after completing the final training requirements.	The system provides a progress page icon in the navigation bar to turn the trainee to the progress page interface and view his/her progress including the certifications after checking if the trainee is completed the final training requirements.	Pass
T9	Allow the trainers to get recommendations and allows the manual trainers search as well.	The system generates trainers' recommendations to the specified trainee based on trainers rating and on the matching place and interests. And allows the manual trainers search and retrieve.	Pass

Table 5.2 System Testing.

The table below shows how we tested the application including screenshots and a brief description.

Chapter 5: System Implementation

TestID	T1	T2	T2
Screenshot			
Description	If the user did not choose either to be a trainer or trainee.	If the user did not field his/her name.	If the user did not choose his/her age.
TestID	T2	T2	T2
Screenshot			
Description	If the user did not choose his/her country.	If the user did not choose his/her city.	If the user did not field a valid email e.g abs@gmail.com.
TestID	T3	T4	T5

Screenshot			
Description	If the user entered less than 7 characters.	When the system authenticates the user using Facebook authentication.	If the trainer wants to edit his/her personal page.
TestID	T6	T7	T8
Screenshot			
Description	If the user wants to view his/her progress and certifications.	If the trainee want to have a real-time chat with the trainers.	If the trainer wants to display the requested list
TestID	T9		

Screenshot		
Description	If the user wants to get recommendations from the system.	

Table 5.3 System Testing2.

5.6 Results and Discussion

As a result of our app, a user can register or log into the application using an android or IOS, entering his own profile info either if the user registered as trainer or trainee. Now our system will recommend trainers for the trainee. We fulfilled our objective we studied the mathematics and algorithms of the recommender system and distinguished the differences between its approaches. We identified which approach gives the closest match according to what our system needs and we chose the right one to our system which content based. We developed a mobile application that helps managing the training process and helps facilitating the complexity of choosing the ideal trainer by giving the best recommendations. We have a clear future plans for the app to make more efficient and easier to use.

5.7 Summary

In this chapter, we discussed how Judan application is implemented, starting from the abstract design to having a full stack application that passes the testing criteria. The process of implementation went through several phases. Starting with searching about the tools and technologies to use, then building the front end using React Native, and after finishing the front end we started building the API server using ASP .NET while connecting the server to SQL database. The final phase was testing the use of our application and we made sure it passed the criteria.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

In conclusion, many people face difficulties finding a superior trainer for a specific hobby or being trainers themselves. For trainees, these problems vary between the shortage of trainers, reaching the training place, the reliability and qualification of trainers, the high prices, and the lack of encouragement. For trainers, the problems are usually focused on the training platform and self-marketing strategies. Our work aims to help finding the most trusted and qualified trainer with the best favorable price for any requested talent, whether a close trainer to meet in person or online, and providing encouragement features for trainees to keep improving in their track. It also aims to provide a well-built platform for trainers with self-marketing features to attract clients. The process of finding the most suitable and matchable trainer is usually tedious and time/money consuming for trainees. The recommender system approach is used to suggest trainers' recommendations customized explicitly for each trainee, which reduces the complexity of the process and increases efficiency. We assume that local talents will improve using Judan application's help, which fulfills a significant need for Saudi 2030 vision. The main work objectives are understanding the problems and needs well, intensively studying the system and algorithms to be used, and effectively finishing the developmental phases.

6.2 Goals Achieved

In this semester, we finished building the basic preparations and most fundamental phases for our training management application, which were mentioned before as objectives. The goals achieved where spread among the chapters as following:

- Problem definition and needs determination.
- Extensive search about the recommender system approach as a background.
- Related work and similar applications analyzing.
- Survey publishing and analysis.
- Requirements elicitation and specification.
- Developmental methodology detailing.

- Architecture, object oriented, and user interface design.
- Data modelling.
- Mapping the design to implementation.
- Full Stack implementation of the application.
- Passing the test of the implementation and having the desired result.

6.3 Limitations and Future Work

During our work and research for this work, we stand up to some limitations and potential improvements that would add more value and increase efficiency. To improve the recommender system functionality, especially the collaborative filtering technique, we need to have the experience and impression of many users to overcome the cold start problem or simply gather their opinion, whether they meet their preferences or not. Another limitation can occur in system supplements, which is a shortage of trainers in the database as a beginning. For example, a trainee might search for a piano trainer at the beginning of releasing Judan application before any registration of a piano trainer in the same area. Adding the feature of online payment needs extra permissions from third parties but is considered necessary. Finally, ensure trainers' efficiency is only done by viewing the rating and trainee's reviews, and that might not be enough.

6.4 Our Contribution:

- We created our own database and dataset.
- We built our own API using asp.net.
- We used two ways for authentication the first one using Facebook authentication and Encryption password (which means that the password will be saved in the database in an encrypted form) for high security.
- We created our own algorithm for the recommender system.

References

- [1] AYTM. HOBBIES Survey. Viewed 12 Oct 2020
<<https://aytm.com/surveys/381582/stat/de9bc373b82855cfcd83b598597dafbd#charts>>. (2016).
- [2] Aggarwal, C. C. Recommender systems (Vol. 1). Cham: Springer International Publishing. (2016).
- [3] Jain, S., Grover, A., Thakur, P. S., & Choudhary, S. K. Trends, problems and solutions of the recommender system. In International conference on computing, communication & automation (pp. 955-958). (2015).
- [4] Mansur, F., Patel, V., & Patel, M. A review on recommender systems. International Conference on Innovations in Information Embedded and Communication Systems (pp. 1-6). (2017).
- [5] Yang, Z., Wu, B., Zheng, K., Wang, X., & Lei, L. A survey of collaborative filtering-based recommender systems for mobile internet applications (pp 3273-3287). (2016).
- [6] Abbattista, F., Degennaris, M., Fanizzi, N., Licchelli, O., Lops, P., Semeraro, G., & Zambetta, F. Learning user profiles for content-based filtering in e-commerce. In Proceedings AI* AI Workshop su Apprendimento Automatico: Metodi e Applicazioni. (2002).
- [7] Ricci, F., Rokach, L., & Shapira, B. Recommender systems: introduction and challenges. In Recommender systems handbook (pp. 1-34). (2015).
- [8] Do, M. P. T., Nguyen, D. V., & Nguyen, L. Model-based approach for collaborative filtering. In 6th International Conference on Information Technology for Education. (2010).

References

- [9] Su, X., & Khoshgoftaar, T. M. A survey of collaborative filtering techniques. *Advances in artificial intelligence*. (2009).
- [10] Su, X., & Khoshgoftaar, T. M. Collaborative filtering for multi-class data using belief nets algorithms. *IEEE international conference on Tools with Artificial Intelligence* (pp. 497-504). (2006).
- [11] Zhang, L., Luo, T., Zhang, F., & Wu, Y. A recommendation model based on deep neural network (pp. 9454-9463). (2018).
- [12] Sohlberg, H. Recommending new items to customers: A comparison between Collaborative Filtering and Association Rule Mining. (2015).
- [13] Gras, B., Brun, A., & Boyer, A. Identifying grey sheep users in collaborative filtering: a distribution-based technique. In *Proceedings of the Conference on User Modeling Adaptation and Personalization* (pp. 17-26). (2016),.
- [14] Thamer, A. Tuthive (Version 1.1), GIS International Technology LLC. Viewed 30 Sept 2020. <<https://apps.apple.com/ae/app/tuthive/id1481101442>>. (2019).
- [15] QuickTutor, LLC. QuickTutor (Version 2.9.0). Viewed 30 Sept 2020. <<https://apps.apple.com/us/app/quicktutor/id1388092698>>. (2019).
- [16] Mohammed, A. HalaPro (Version 2.0). Viewed 30 Sept 2020. <<https://apps.apple.com/us/app/halapro/id1440117615?ls=1>>. (2018).
- [17] BOK Solutions, Inc. Gooroo (Version 1.5). Viewed 30 Sept 2020. <<https://apps.apple.com/us/app/gooroo-for-students/id1482457098?mt=8>>. (2019).
- [18] Sommerville, I. Software engineering. Pearson Education India. (2004).
- [19] Paul, A., & Nalwaya, A. React Native for Mobile Development (pp. 14-256). (2019).



[20] Novick, Vladimir. React Native-Building Mobile Apps with JavaScript. Packt Publishing Ltd, 2017.

Appendix A: Survey

Conducted survey questions that were used to gather information and obtain project requirements.

استبيان التدريب الشخصي

استبيان لمشروع تخرج طالبات علوم حاسوب في جامعة طيبة 2020-2021

:::

العمر

* من 9 - 12

من 13 - 15

من 16 - 19

من 20 - 29

من 30 - 39

من 40 - 59

من 60 - 74

من 40 - 59

من 60 - 74

* الجنس

أنثى

ذكر

⋮⋮⋮

* الجنسية

سعودي/ة

غير سعودي/ة

:::

*** المدينة**

1. الرياض

2. مكة المكرمة

3. المدينة المنورة

4. جدة

5. الطائف

6. تبوك

7. الدمام

8. الأحساء

9. القصيم

10. حائل

11. الباحة

12. عسير

13. جيزان

14. نجران

15. الخبر

16. الجوف

17. الجبيل

18. القطيف

19. أبها

20. ينبع

21. العلا

22. الحناكية

23. بدر

24. غير ما ذكر

* هل أنت موظف

نعم

لا

...

* المستوى التعليمي

ثانوي

بكالوريوس

ماجستير

دكتوراة

غير ذلك...

* هوايتك (قد لا تبرع في القيام بها لكنك تستمتع بمارسها)

نص الإجابة القصير

:::

* موهبتك أو مهارتك (تبرع في القيام بها)

نص الإجابة القصير

* هل ستكون مهتماً في حال أتيحت لك فرصة تدريب المهتمين على
موهبتك (حضورياً أو عن بعد) مقابل مبلغ مادي مناسب؟

نعم

لا

ربما

* إذا كنت تفكّر في الاستفادة مادياً من موهبتك سواءً كمقدم لها أو كمدرب، هل تتوقع أن تواجه صعوبة في التسويق لذاتك وجذب المهتمين؟

نعم

لا

ربما

* بالنسبة لهوايتك التي تستمتع بها، هل تتوقع أن تتطور بشكل أسرع في حال تواجد من يُدرب عليها شخصياً ويُتابع تقدمك فيها؟

نعم

لا

ربما

* هل واجهتك صعوبة معينة في إيجاد مدرب جيد لتطوير هوايتك؟

نعم

لا

إذا كانت الإجابة "نعم"، فضلاً اذكر هذه الصعوبة

نص الإجابة الطويلة

* هل تعتقد أن هناك من يُدرب الآخرين مقابل مبلغ مادي دون أن تكون لديه خبرة كافية؟

نعم

لا

ربما

:::

* هل تعتقد أن أسعار المدربين (بغض النظر عن مجالهم) مرتفعة في أغلب الأحيان؟

نعم أعتقد ذلك

في بعض الأحيان فقط

لا أعتقد ذلك

لا أعلم

* هل يمكنك الاتفاق مع شخص على موقع التواصل أو غيره لغرض التدريب أم أنه تخشى ذلك لضعف الثقة

نعم يمكنني الاتفاق

أتجنب ذلك بسبب ضعف الثقة

لست متأكداً

* هل سبق وأن رغبت في التدريب الشخصي وجهًا لوجه بغرض
تنمية هواية ووجدت صعوبة في إيجاد مدرب قرير؟

نعم

لم يسبق

ربما

* هل سبق وأن تعذر عليك الذهاب لمقر تدريبك المعتاد بسبب جائحة
كورونا؟

نعم

لا ينطبق

⋮⋮⋮

* هل تفتقد عنصر التشجيع على هوايتك أو موهبتك؟ *

نعم

لا

ربما

* هل تؤيد استغلال مواهب الشعب السعودي وإشراكهم في تقديم الفعاليات الترفيهية؟

نعم

لا

Appendix B: React Native

React Native is an open-source platform with one of the largest support communities. Unlike typical hybrid applications, this new platform is designed for mobile platforms. Based on a JavaScript library, React Native allows you to create a single JavaScript code base that will run on multiple mobile devices (iOS, Android and Windows). Therefore, before moving on, it is important to understand why cross-platform development is becoming more popular. A review to the rise of native applications. It all started in 2012 with the deep regret of Facebook CEO Mark Zuckerberg (Mark Zuckerberg), who said: "As a company, the biggest mistake we make is to invest too much in HTML5 instead of native.". As promised, the company launched the react-native framework in 2015. In the initial version, the company can only develop applications for the iOS platform. But gradually, the company has also expanded support for Android native applications. The difference in popularity between react native and all other frameworks is huge. Here is a glimpse according to Google Trends:

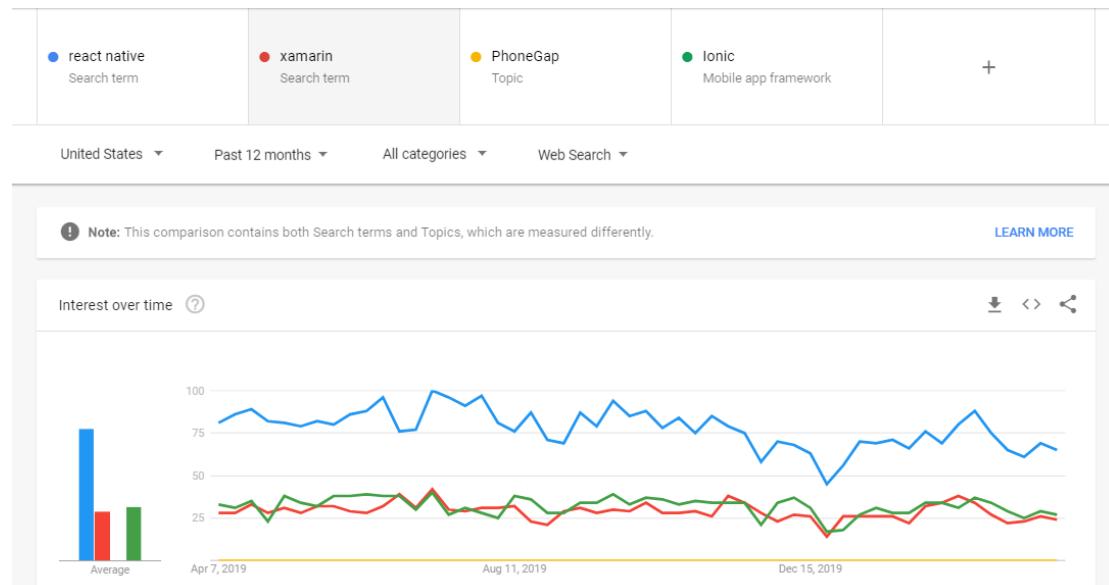


Figure 5.2 A glimpse according to Google Trends. [20]

Cross-Platform Development

The difficulty of developing back-ends suitable for different operating systems brought up the idea of hybrid applications. Another option is to develop separate code for

Appendix B: Survey

different operating systems. But this is a time-consuming and expensive method. Therefore, a large part of entrepreneurs started to use hybrid applications. Though React Native was originally introduced on iOS. With amazing capabilities and features, Facebook has built support for Android. Due to the greater advancement in the framework, it is now easy to embed native code for platform-specific authenticity. This further reduces the cost and development time to a considerable amount. Benefits of using Cross-Platform Approach:

- It is a cost-effective solution
- The code is reusable
- Easy access to plugins
- Rapid development
- Ideal for prototyping

The above advantages of cross-platform make it an ideal approach to developing mobile applications. However, it also leads to confusion. This is because there are many platforms available that offer a hassle-free development process. What do you think the user can expect from the mobile application? The answer is perfect, personalized experience and real-time information. Today, it is not easy to impress users, but it is possible. All you need is performance-driven and cost-effective mobile applications. Therefore, startups will consider using this programming language and prioritize hiring native developers. React Native apps provide the same seamless user experience and performance as native apps. In addition, it is easy to learn and allows reactive programmers to release updates immediately without having to enter the app store. In fact, its modular & intuitive interface makes it easy for the developers to transform web apps into mobile apps. For all the above reasons, it is also an ideal solution for the development of MVP. Without spending a lot of money, you can get your mobile app running within a short time span. [20]