Koobecaf.

Grupa B5

June 14, 2023

Abstract

Koobecaf is an innovative web application that aims to redefine digital connectivity and engagement. The platform offers a comprehensive range of features, including a powerful messenger for seamless communication and personalized advertisements based on users' interests. Through the messenger feature, users can engage in real-time conversations, fostering meaningful connections. The platform also provides targeted advertisements that align with individual preferences, enhancing the user experience.With a commitment to user satisfaction, Koobecaf provides a dynamic and personalized platform for enhanced connectivity and engagement.

1 Introduction

In an increasingly interconnected world, the ability to connect, share, and engage with others has become paramount. Web applications have played a significant role in facilitating these interactions, enabling individuals to communicate, express themselves, and discover new content. This project documentation presents Koobecaf, an innovative web application that aims to revolutionize digital connectivity and engagement by offering a comprehensive range of features. Our platform goes beyond traditional social networking by integrating a powerful messenger and personalized advertisements based on users' interests, providing a dynamic and personalized user experience. We chose the light purple and white for the messages' bubbles as purple combines the calm stability of blue and the fierce energy of red and white means wholeness and completion all of these being representative for our team.

2 Proposed solutions

The primary objective of Koobecaf is to create an environment where individuals can establish meaningful connections with others. By leveraging advanced algorithms and network analysis techniques, the application aims to enhance the process of finding and connecting with like-minded individuals. Through intuitive tools and features, users can effortlessly discover and connect with individuals who share similar interests, backgrounds, and experiences.

Koobecaf recognizes the significance of seamless communication in fostering meaningful connections. Our integrated messenger feature allows users to engage in real-time conversations, facilitating instant communication and enabling users to stay connected with friends, family, and colleagues. In addition to fostering connections, Koobecaf takes personalization to the next level by offering targeted advertisements based on users' interests. By leveraging advanced algorithms and data analysis, we curate advertisements that align with individual preferences .This personalized approach ensures that users are presented with relevant and engaging advertisements, enhancing their overall experience while enabling businesses to reach their target audience more effectively.

The development of Koobecaf has been guided by user-centric design principles, focusing on creating an intuitive and seamless interface that maximizes user engagement and satisfaction. Through rigorous testing and optimization, we have created a platform that delivers a reliable and enjoyable user experience across a wide range of devices.

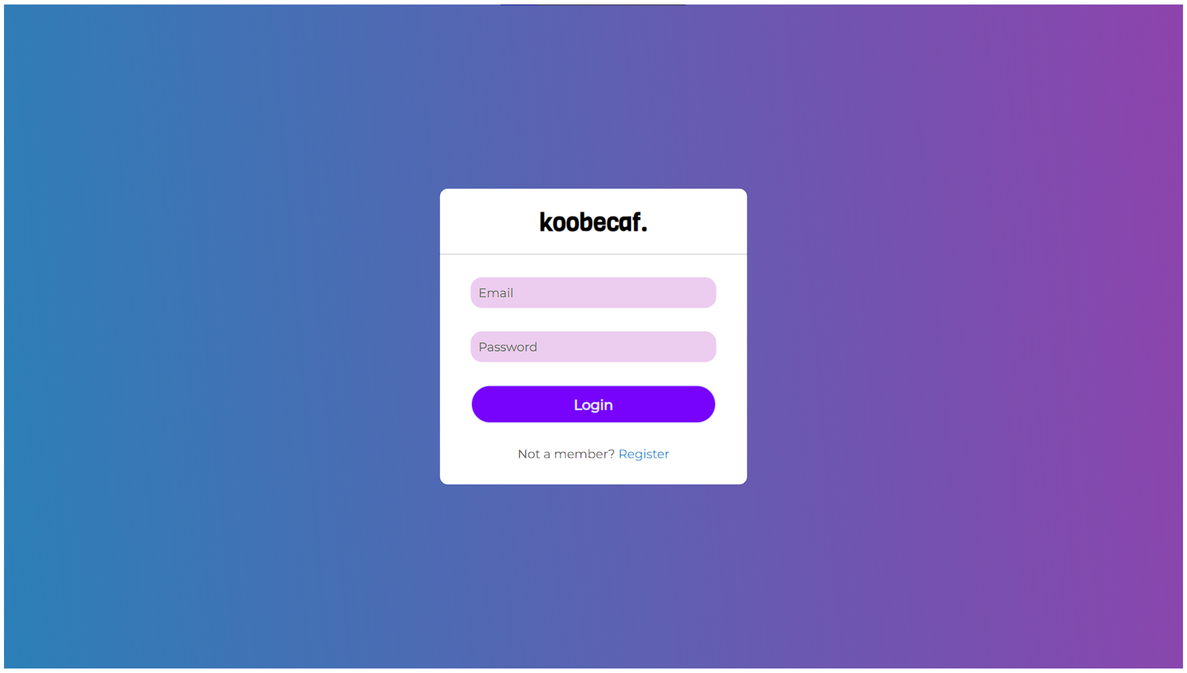
3 Arhitecture

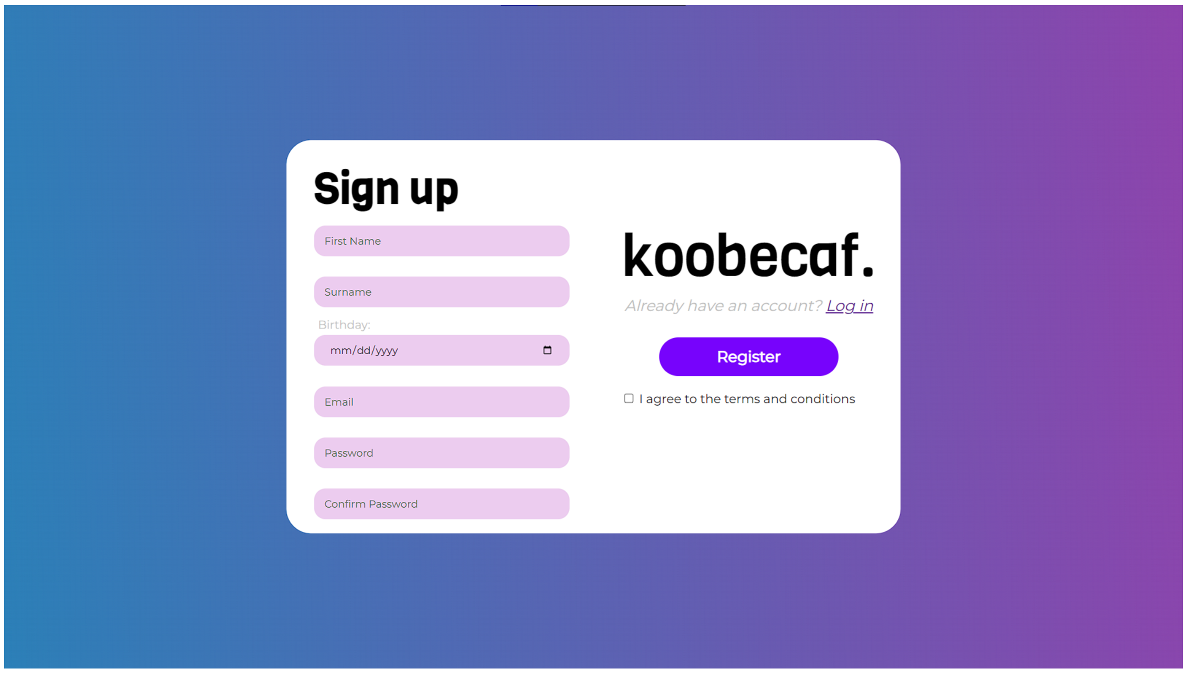
3.1 Frontend

The Frontend module is written in React to create modular and reusable code that can help us scale our web application in the future. The user will interact with several beautifully designed pages outlined by our team of curious and hardly working frontend team. We chose the purple as our primary color because it is often associated with royalty, nobility, luxury, power, and ambition.

3.1.1 Login and Register

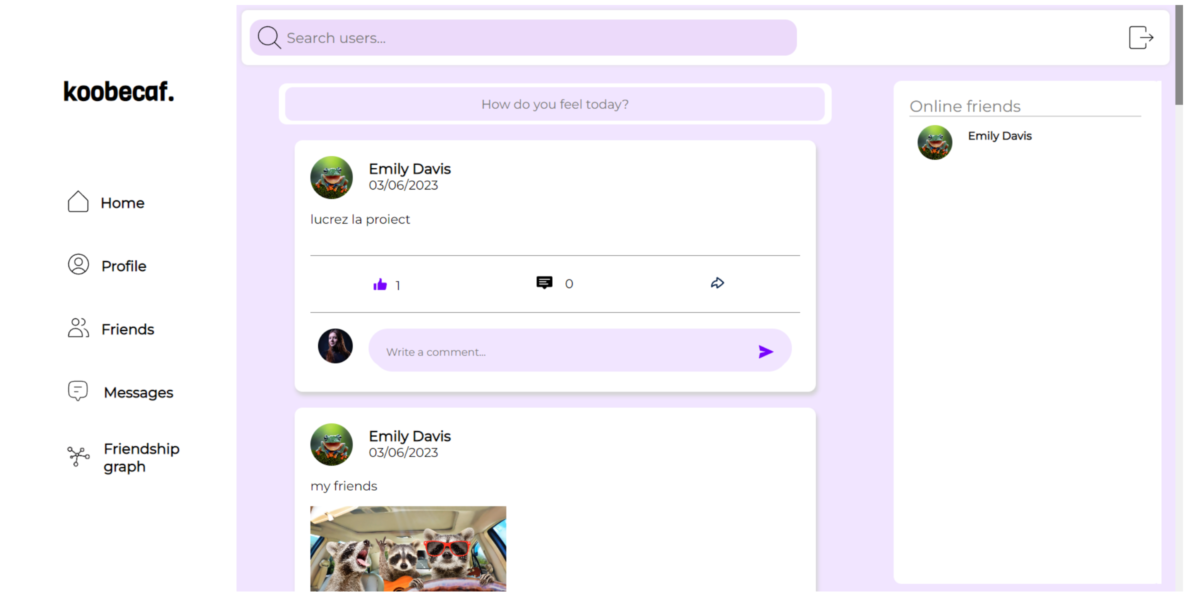
These are the main 2 pages that will greet user the first time they access the website, depending on whether they already have an account or not. Both of these are fairly straightforward to use. The user is required to fill in the fields with the necessary and/or correct data and click *Register* or *Login*. Upon creating an account or simply login in with an already existing one, the user will be redirected to the feed.





3.1.2 Homepage

This is the first page one sees immediately after logging on the web app. On the left, it features 5 main options that will redirect the user to other functionalities of the websites. On the right, it will always display friends that are currently online or were logged off only recently. The main purpose of this page is to allow a registered user to see and interact with posts, through likes and comments. The user can also make his own posts. It's to be noted that this is the same page ads will be displayed regularly, according to the suggestion algorithm. The ads are displayed in a similar fashion as the user-made posts, except they have a little tag indicating that it's an advertisement, to avoid problems with the law.

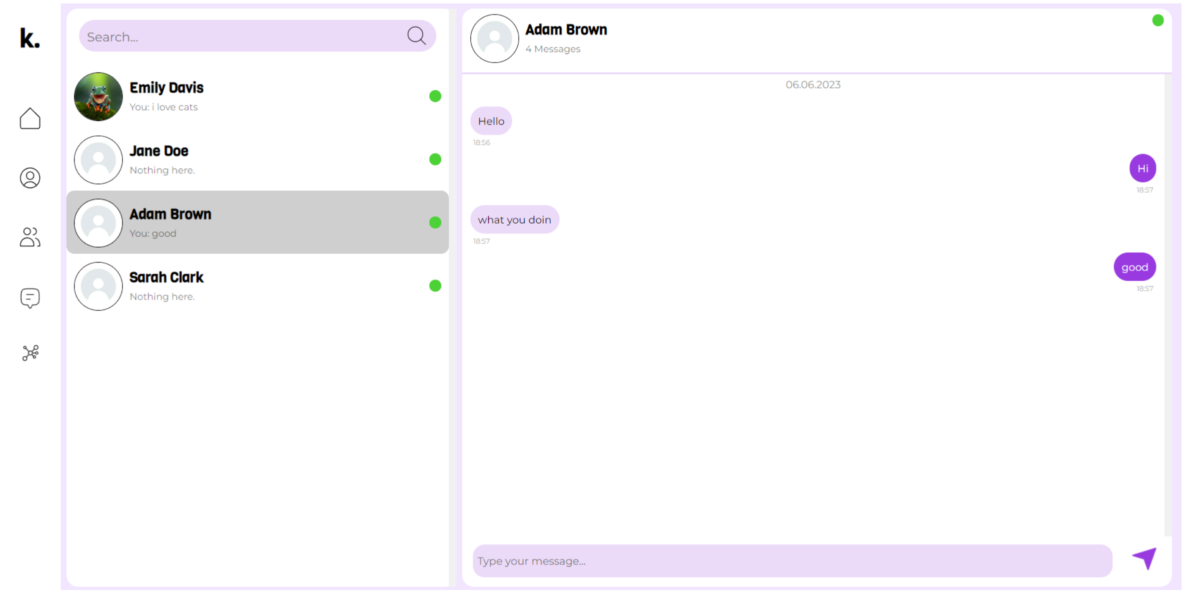


Emily Davis
03/06/2023
lucrez la proiect

Emily Davis
03/06/2023
my friends

3.1.3 Messages

This is one of the key pages the user will interact with since it provides real time chatting with your friends. On the far left we have the same menu as the home feed, with buttons to quickly access the other pages. A search bar is present at the top to allow for searching of a specific person, in the case their chat isn't among the most recent ones, that are listed below the search bar. One can know whether the person they want to talk to is online through the dot on the right of their chat card, colored green when he's on and red otherwise. Whenever you send or get a new message, the conversation auto scrolls to be up to date. Lastly, if you were to receive a message from a chat you don't have opened at the moment, you'll have a notification on said conversation card to indicate you have unread messages.



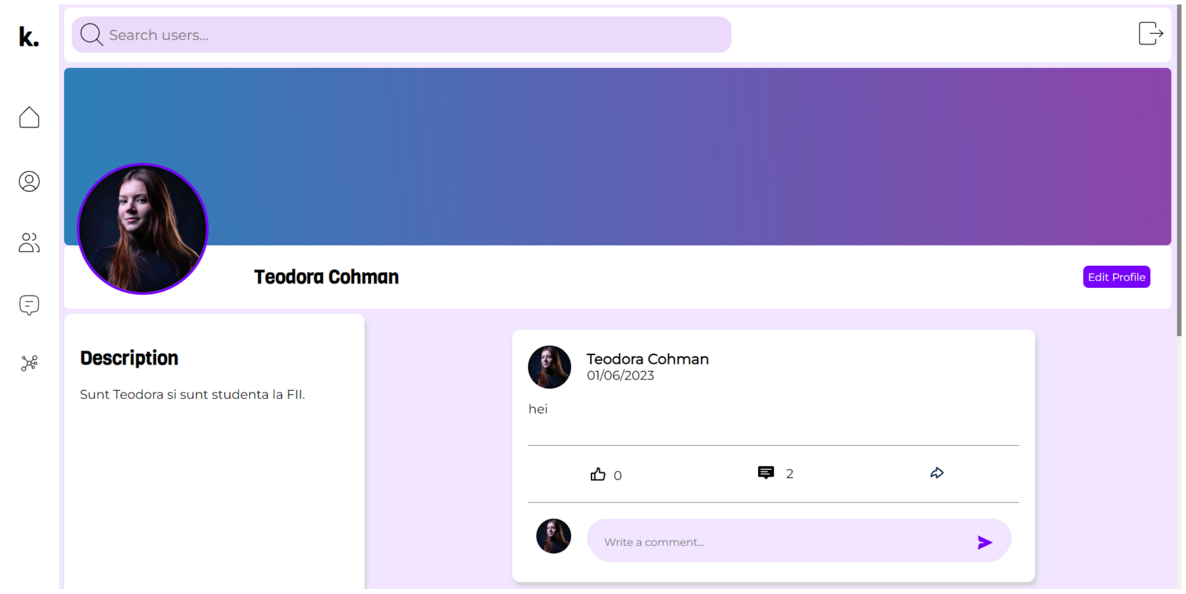
Emily Davis
You: i love cats
Jane Doe
Nothing here.
Adam Brown
You: good
Sarah Clark
Nothing here.

Adam Brown
4 Messages

06.06.2023
Hello
18:56
Hi
18:57
what you doin
18:57
good
18:57

3.1.4 Profile

This is the page showcased when a user accesses its own profile or the profile of a friend. It's main perks are displaying any sort of description, profile picture and cover picture set by the account owner. It also features the full timeline of said user, by scrolling down indefinitely till it reaches the very first post created. When accessing your own account, you have the option to change your profile or cover picture by simply clicking on them and choosing upload.



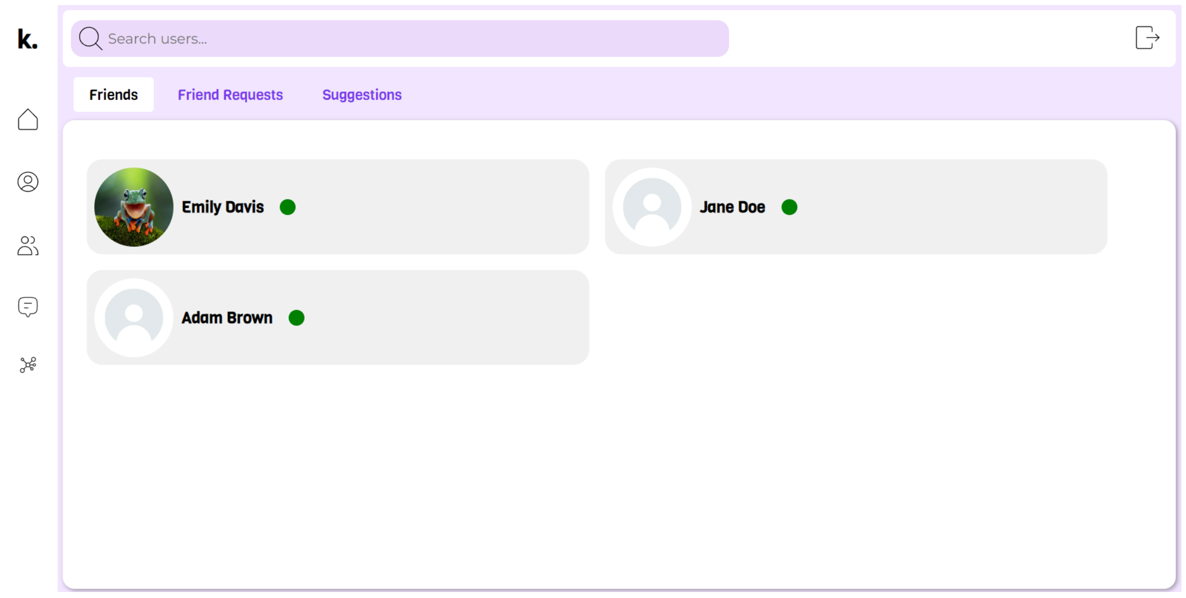
Teodora Cohman

Description
Sunt Teodora si sunt studenta la FII.

Teodora Cohman
01/06/2023
hei
0
2

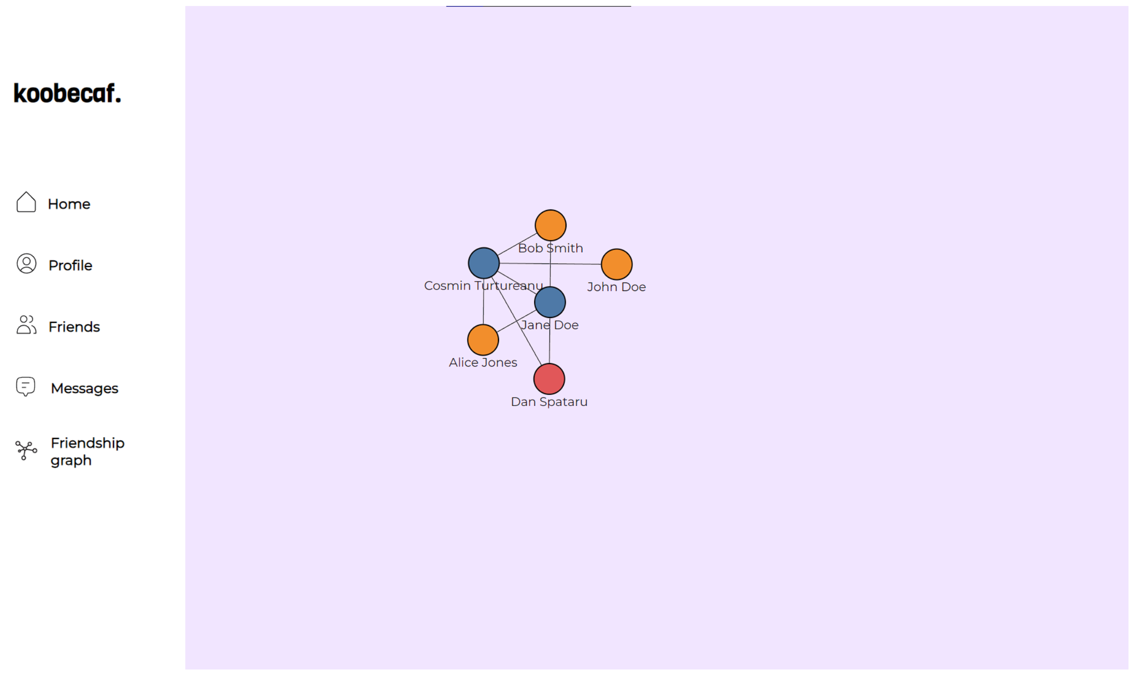
3.1.5 Friends

This page is split into 3 main sections. First is your current friend list, with both online and offline users, from where you're able to modify your relationship with them or simply go to their profile. The second one is the *friend suggestions* section, where the app may recommend to you to add *friends-of-friends* to your own friend list. In the case of having a fresh account with no friends, the algorithm would initially recommend random users. Lastly, we have the *friend requests* section, where all notifications regarding people who wish to be your friend on our platform will be showcased, giving you the option to accept or deny their request. Ignoring them will leave the status of the request as *pending*.



3.1.6 Friendship Graph

A standout feature of our application is the ability to see a visual representation of your social life through a graph-like structure, on this specific page. The graph features different colors and vertices to represent who is your current friend, who could be suggested to you as a friend, and who are, for now, strangers to you. The visual representation is animated and can be interacted via the cursor, although that serves no practical purpose for now, beyond aesthetics.



3.2 Backend

The backend is written in Java and it uses the Springboot framework to fulfill the necessary requirements. It consists of 3 modules: **Core**, **Conversations** and **Ads**. The only module needed by *Koobecaf.* is the **Core** module, as the other ones are treated as plug-ins.

3.2.1 Core

This is the main module of the *Koobecaf.* and it handles the interaction between users, using posts, comments and likes. The endpoints exposed are provided through a set of controllers, the most important ones being:

- **AdminController** - exposes endpoints required for constructing the friendships graph of a user. It retrieves from the database, the information about the friendships between users and sends them to the front-end in the form of a pair between a list of links (a link is a tuple consisting of two user ids) and a list of users.
- **FileController** - uses the *Cloudflare* service in order to upload, download and delete files from the user. The files could be anything from images and videos to PDFs and other types of files. In the uploading process, it firstly receives the file to be uploaded as a *MultipartFile*, then it makes a call to the *Cloudflare* API. The name used for the uploaded file consists of the base64 encoding of the file's SHA256 hash, concatenated with the extension of the uploaded file. For the downloading and deletion processes, the same name is used.
- **UserController** - from this controller, one endpoint in particular is worth mentioning the most: */suggestions*. This endpoint provides to the user, a list of people that he/she may know. The algorithm behind the recommendation works by looking at the user's "2nd degree" friends, in case they exist. Otherwise, random people are recommended.

3.2.2 Conversations

This module provides users the functionality necessary in order to have a real-time chat, using WebSockets. As dependencies, it uses *spring-websocket* and *spring-messaging*, both part of the SpringBoot framework. Although we have the functionality to have a group chat, it was not implemented yet in the front-end. The users communicate through a channel which is assigned to them when the first message in the conversation is sent. In case the receiving user is not logged on when the sender sends a message, the messages are stored in the database to then be retrieved by the receiving user once they log on.

3.2.3 Ads

This is the module that deals with the provisioning of ads to the user. It uses an ad profile that is associated with a specific user in order to recommend relevant ads to them. For the recommendation part, it analyzes the user generated content (posts, comments, messages) in order to create an ad profile that is as accurate as possible. The user generated content is analyzed using the *edu.stanford.nlp* dependency, which is a module for natural language processing. The content is fed into the *Stanford-CoreNLP* pipeline and then, for each sentence, its sentiment score is deduced (that is, if the user has a positive feeling about the discussed subject or not) and the ad profile is updated with the relevant keywords extracted from the sentences.
Another aspect of the recommendation is the actual provisioning of ads, based on the ad profile created for the user. This is done by comparing the keywords linked to an ad, with the keywords from a user's ad profile by using a function for cosine similarity:

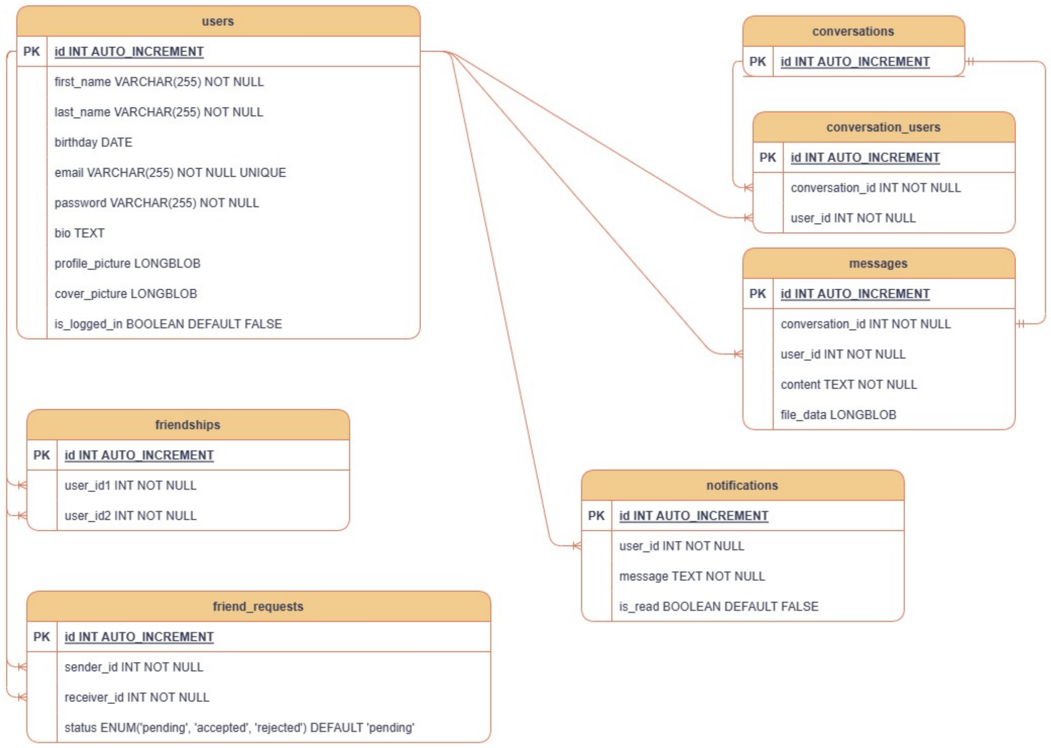
$$S_C(A,B) := cos(\theta) = \frac{A \cdot B}{\| A \| \cdot \| B \|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The user input is automatically sanitized by using the associated JPA repository of the model before being introduced in the database.

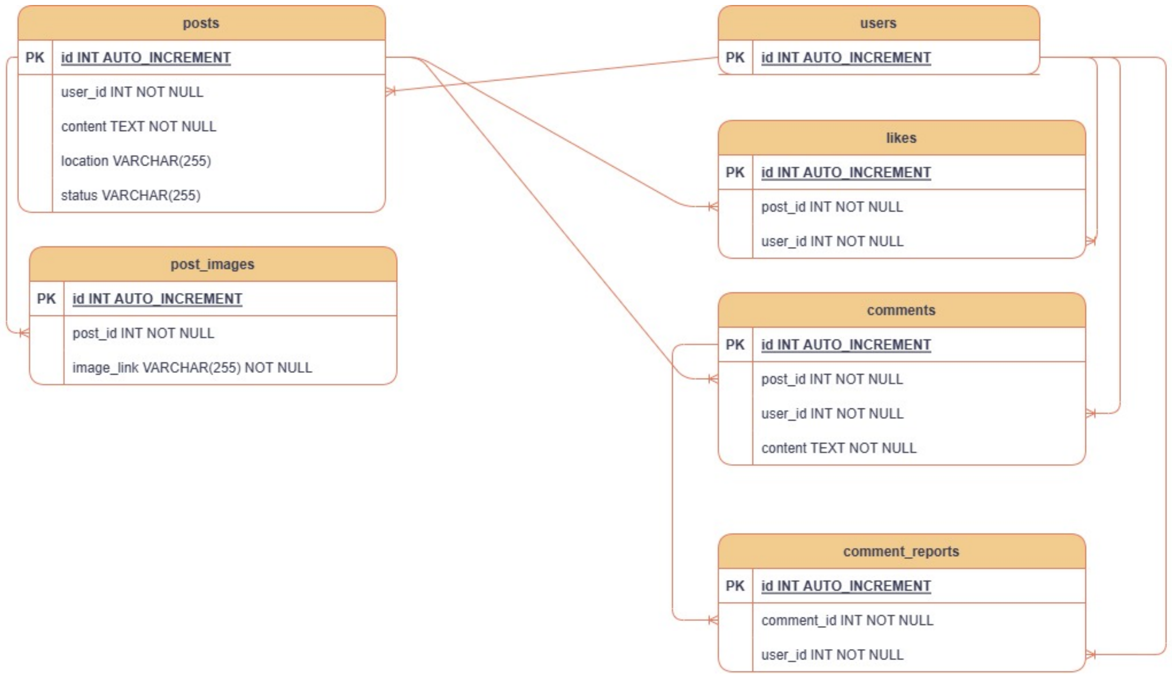
3.3 Database

The **Koobecaf.** application makes use of a relational **MySQL** Database to manage its various data entries and the relationship between them in the context of a social media platform. While it contains most entities expected of such a platform, such as *users*, *posts*, *messages*, it distinguishes itself from others through the additional support tables for data integrity purpose, in case of *many_to_many* relationships, such as users and conversations, or images and posts. We also took some security precautions, such as storing the hash of the password, as the standard dictates, rather than the password itself. It's main distinguishable feature is due to the ads related tables, such as *ad_profiles*, that store data related to topics of interest for each individual user, later to be contrasted against a set of ads stored in the database to facilitate the recommendation functionality. It also features a few triggers to ensure that timestamps of inputs or updates remain consistent, such as in the case of the *conversations* table, where it's *updated_at* field has to be modified each time a new message is sent in that specific conversation. Note that the following diagrams have been split up into 3 modules and had the less important columns (updated_at, created_at) stripped out to avoid clutter and make it easier to read. It's meant to showcase the relationships between tables.

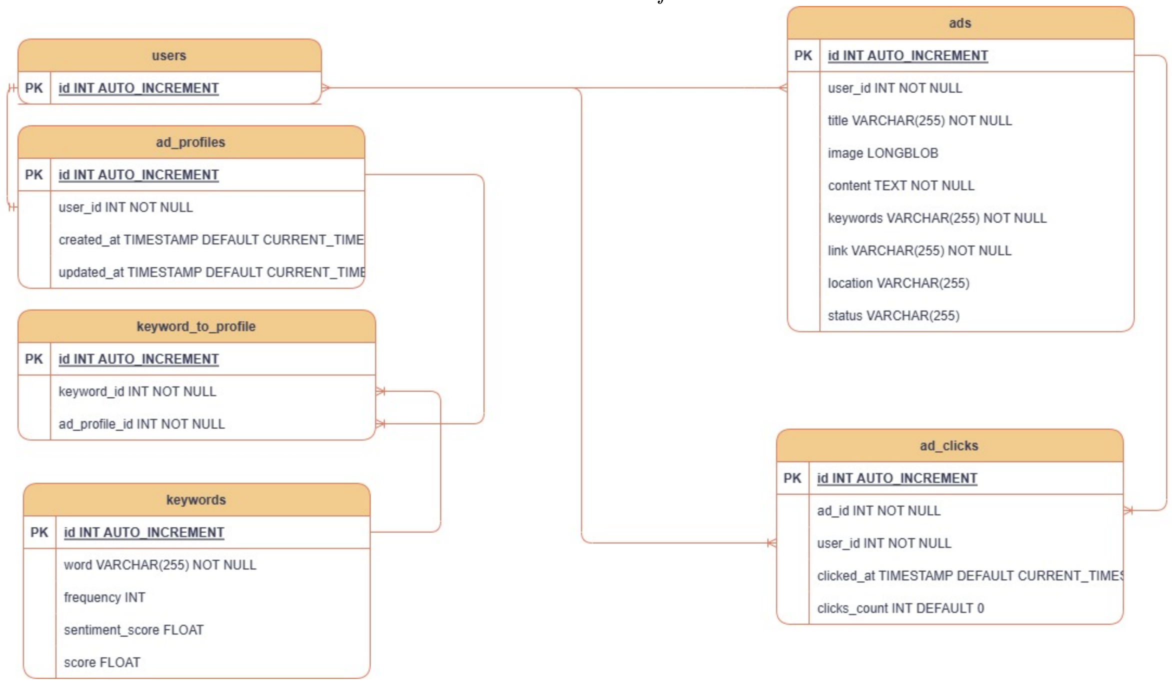
Messages and Friendships



Posts, Likes and Comments



Ads and Ad Profiles



4 Future work

- **Group Messaging** - Make use of the auxiliary *conversation_users* table from the database with it's *many-to-many* relationship to allow for more than 2 users to take part in the same conversation. This would require a degree of optimisation for the database as to be able to handle an increase numbers of messages all with the same *conversation_id*.

- **Multimedia support for Messages/Comments** - Add the option to send user-uploaded images or predetermined emojis/gifs to all forms of messaging, whether it's direct messaging, group chats or comments on posts. It would also allow users to upload videos, not just photos, on their posts.

- **Ad Workshop** - Environment that would advertisement companies or users with their own business to create and manage their own personalized ads. This would require a special type of business oriented account, as a safety precaution against spam and inappropriate content in the newly made ads.
- **Role based access control (RBAC)** - Different accounts will have different roles that will allow the access to particular functionalities. For example, a page for monitoring user statistics that will only be accessible to admins.
- **Localisation** - Offer the option of different language support depending on the user's registered place of living or just personal preference.
- **Smartphone support** - Make the app operate to the same degree and precision as its current desktop version.

5 Conclusions

In conclusion, Koobecaf sets itself apart by integrating a powerful messenger and personalized advertisements based on users' interests. By combining seamless communication and tailored advertising, Koobecaf provides a dynamic and personalized user experience.

References

1. **MySQL:** https://www.mysql.com/
2. **React:** https://react.dev/learn
3. **Spring:** https://spring.io/
4. **Docker:** https://www.docker.com/
5. **Cloudflare Images:** https://developers.cloudflare.com/r2/get-started/
6. **Diagrams Software:** https://www.diagrams.net
7. **Thymeleaf:** https://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html
8. **WebSockets:** https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API/Writing_WebSocket_client_applications