

Universidad Distrital Francisco José de Caldas



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

Technical Report UDinder

Advanced Programming

Authors:

Nicolas Avendaño Barajas - 20231020113

Juan Sebastian Vega Diaz - 20231020087

Teacher:

Carlos Andres Sierra Virgüez

Bogotá D.C.

Description Of The Project

The project involves creating a monolithic software for an online platform. In terms of technical decisions, the project focused on key processes such as sign-in, sign-up, authentication, profile completion, and interaction with other user profiles.

Backend development was carried out using Python, leveraging the FastAPI framework. For the frontend graphical interface, HTML, CSS, and JavaScript were chosen, along with the Apache server and the Django framework.

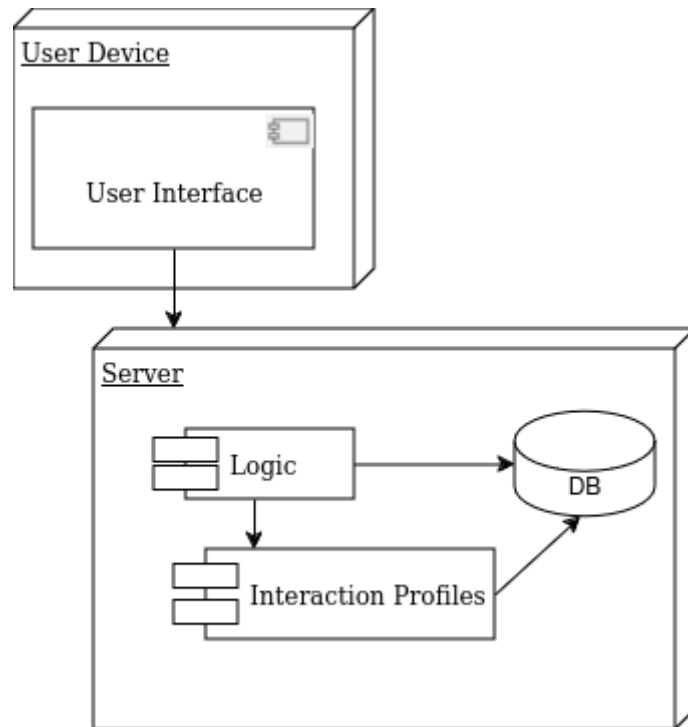
For managing the database, SQLAlchemy was selected due to its ability to handle database relationships with a syntax similar to that of the programming language.

User Stories

1. As a user, I want to be able to sign up for the app so that I can create a profile and start using the platform.
2. As a user, I want to be able to sign in to the app with my username and password so that I can access my profile and interact with other users.
3. As a user, I want to be able to view suggested profiles so that I can find potential matches.
4. As a user, I want to be able to swipe right or left on profiles to indicate my interest or disinterest.
5. As a user, I want to receive notifications when I have a match with another user so that I can start messaging them.
6. As a user, I want to be able to send messages to my matches to start a conversation and get to know them better.
7. As a user, I want to be able to view my matches and their profiles so that I can keep track of my interactions.
8. As an administrator, I want to be able to access the database to view user information and activity for moderation purposes.
9. As an administrator, I want to be able to block or suspend users who violate the app's community guidelines or terms of service.
10. As an administrator, I want to be able to review reported profiles and messages to ensure they comply with the app's policies.
11. As an administrator, I want to be able to respond to user inquiries and provide assistance with any issues they may encounter.
12. As a developer, I want to implement user authentication functionality to allow users to sign up and sign in securely.
13. As a developer, I want to implement profile management features to allow users to update their profiles and preferences.
14. As a developer, I want to implement the matching algorithm to suggest compatible profiles to users.
15. As a developer, I want to implement the messaging functionality to allow users to communicate with their matches.
16. As a developer, I want to implement a reporting system to allow users to report inappropriate behavior or content.
17. As a developer, I want to optimize the app's performance and scalability to handle a large number of users and interactions.
18. As a developer, I want to continuously monitor and improve the app's security to protect user data and privacy.

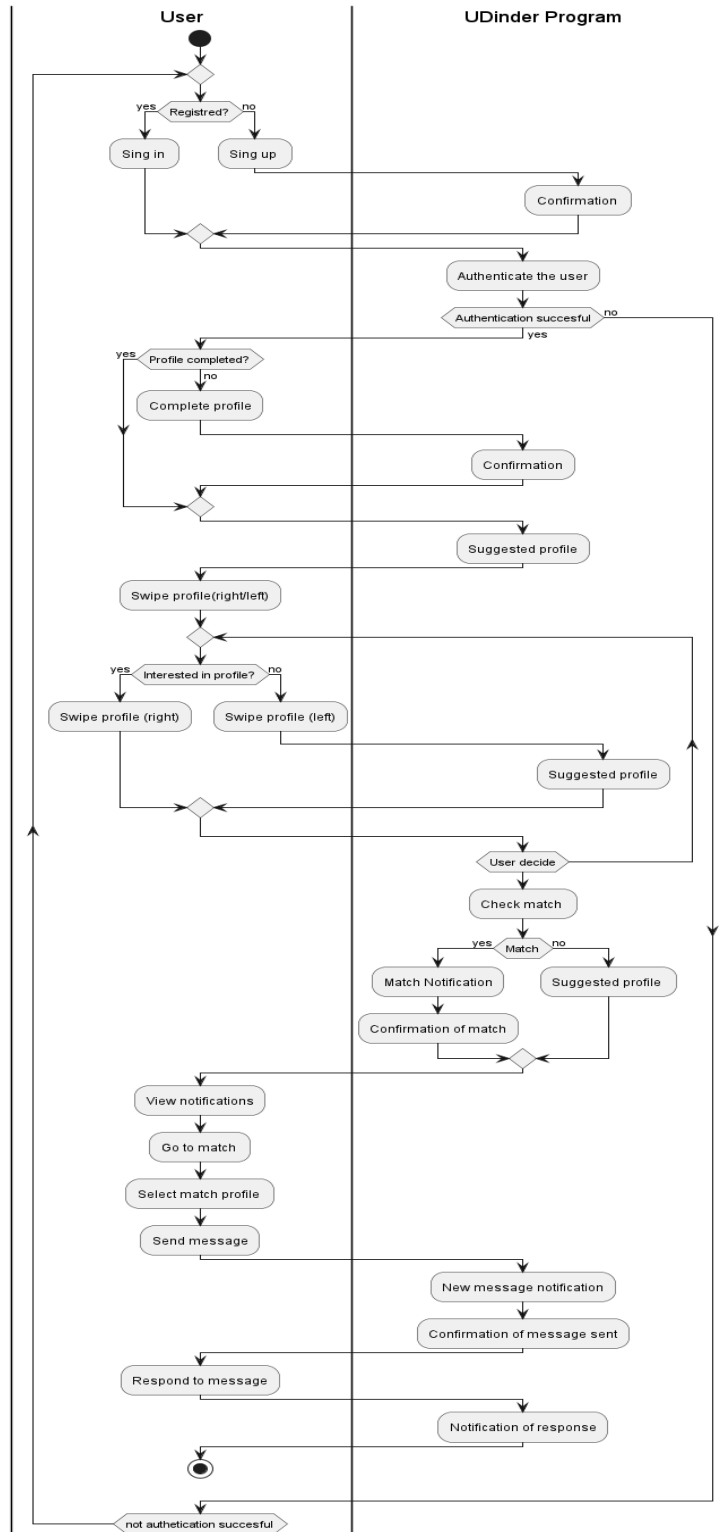
Uml Diagrams

- **Deployment Diagram:**

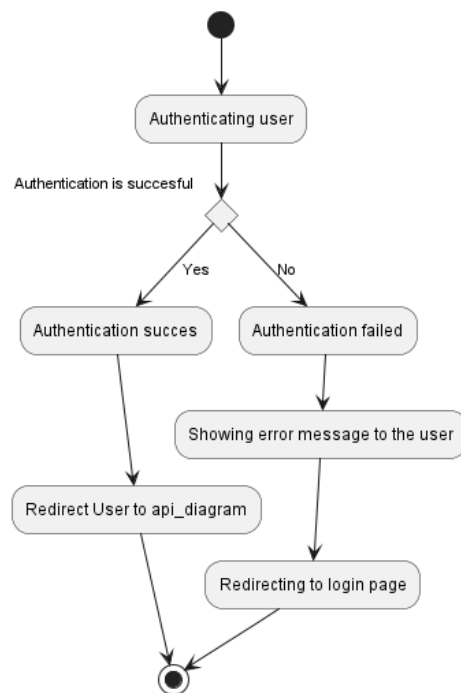


- Activities Diagrams:

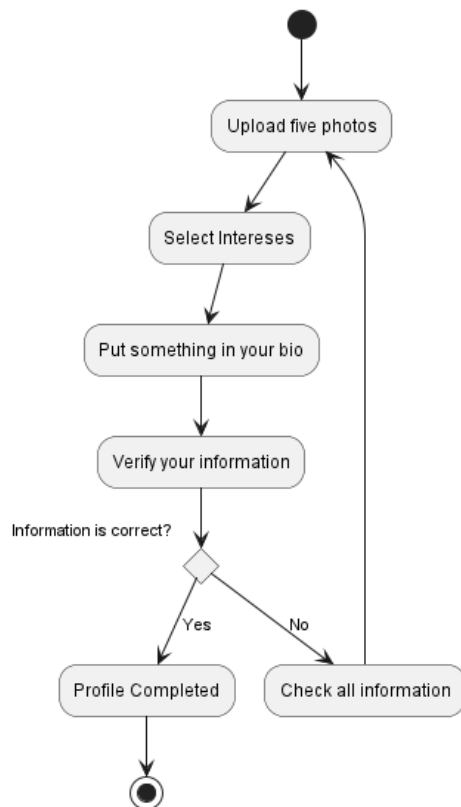
- General_ad:



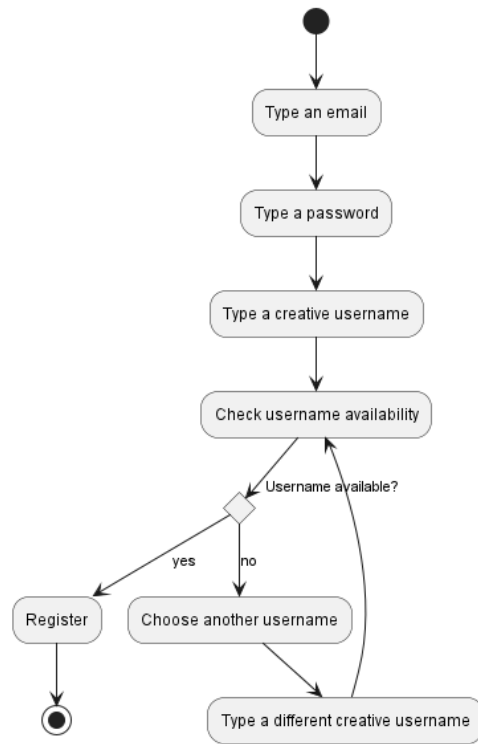
- **Authentication_ad:**



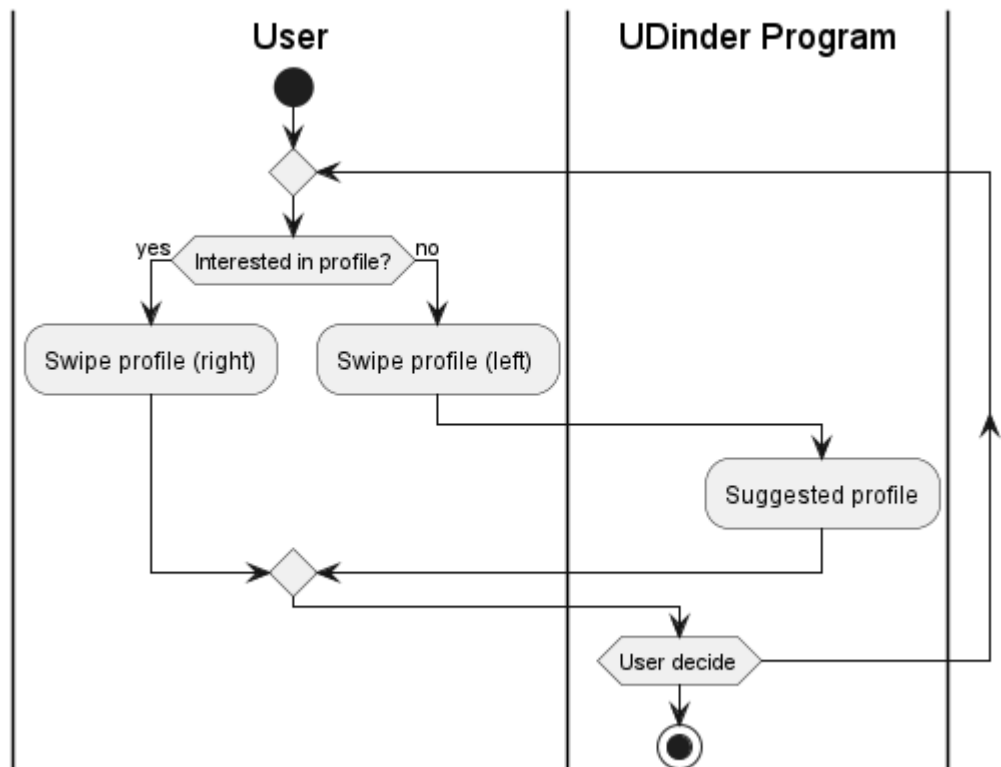
- **Complete_profile_ad:**



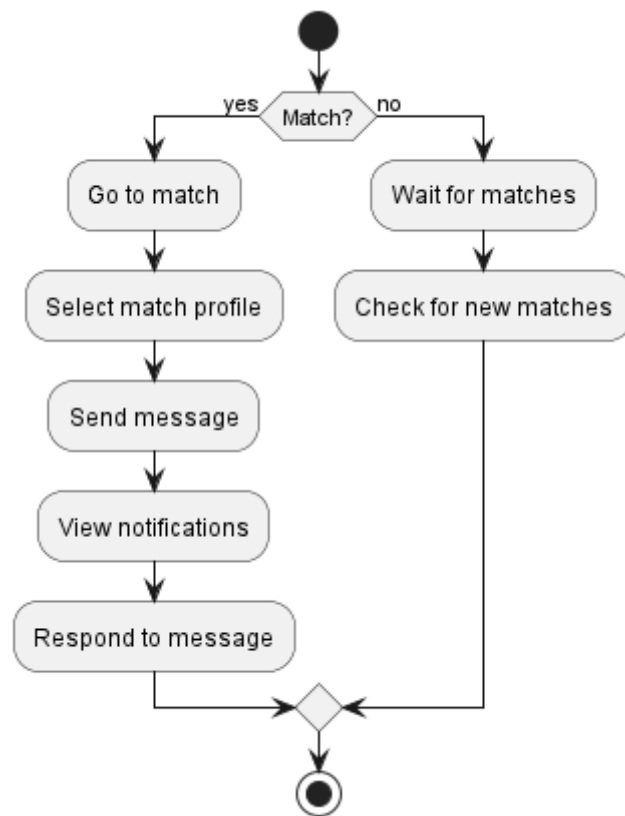
- **Sing_up_ad:**



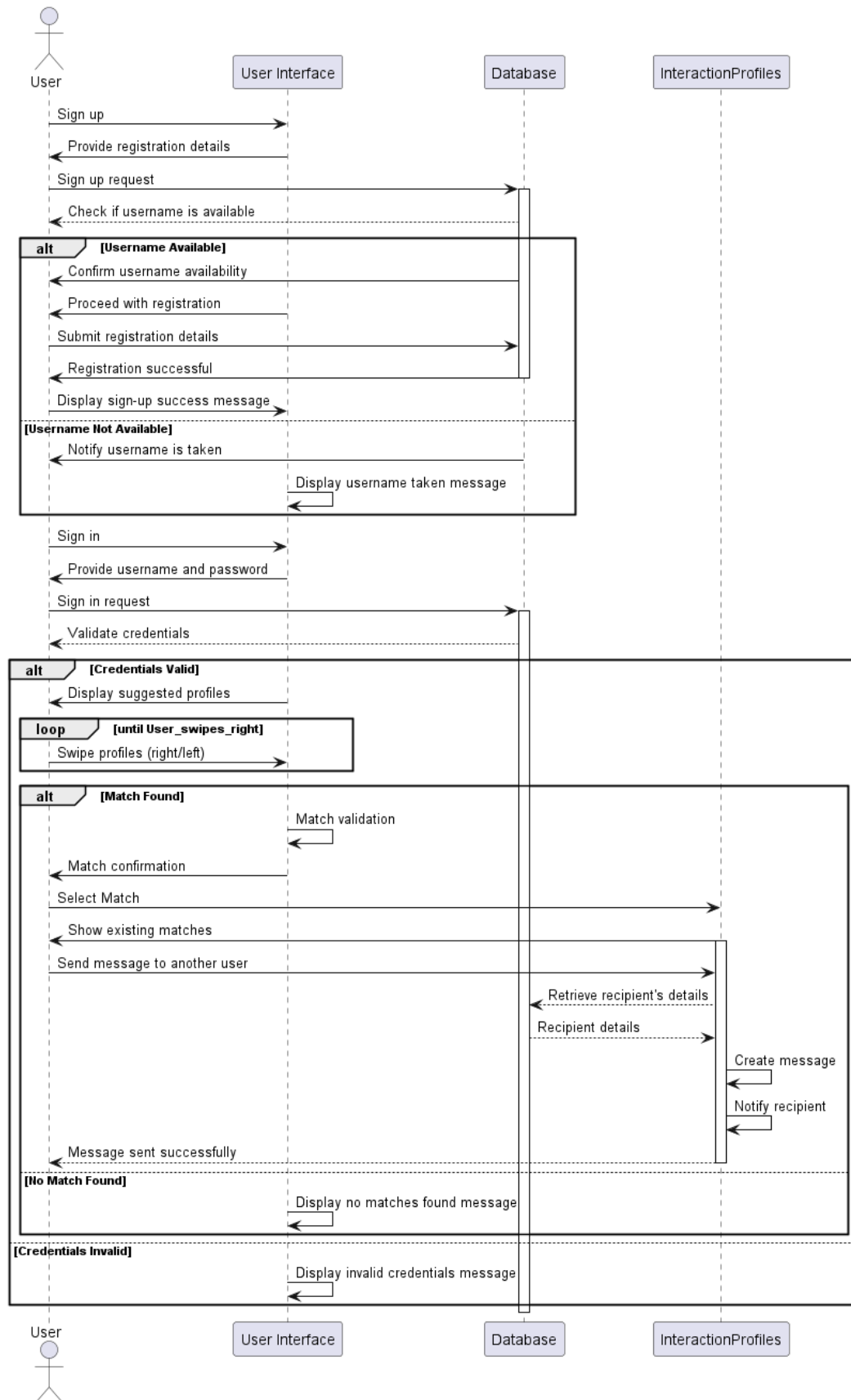
- **Swipe_function_ad:**



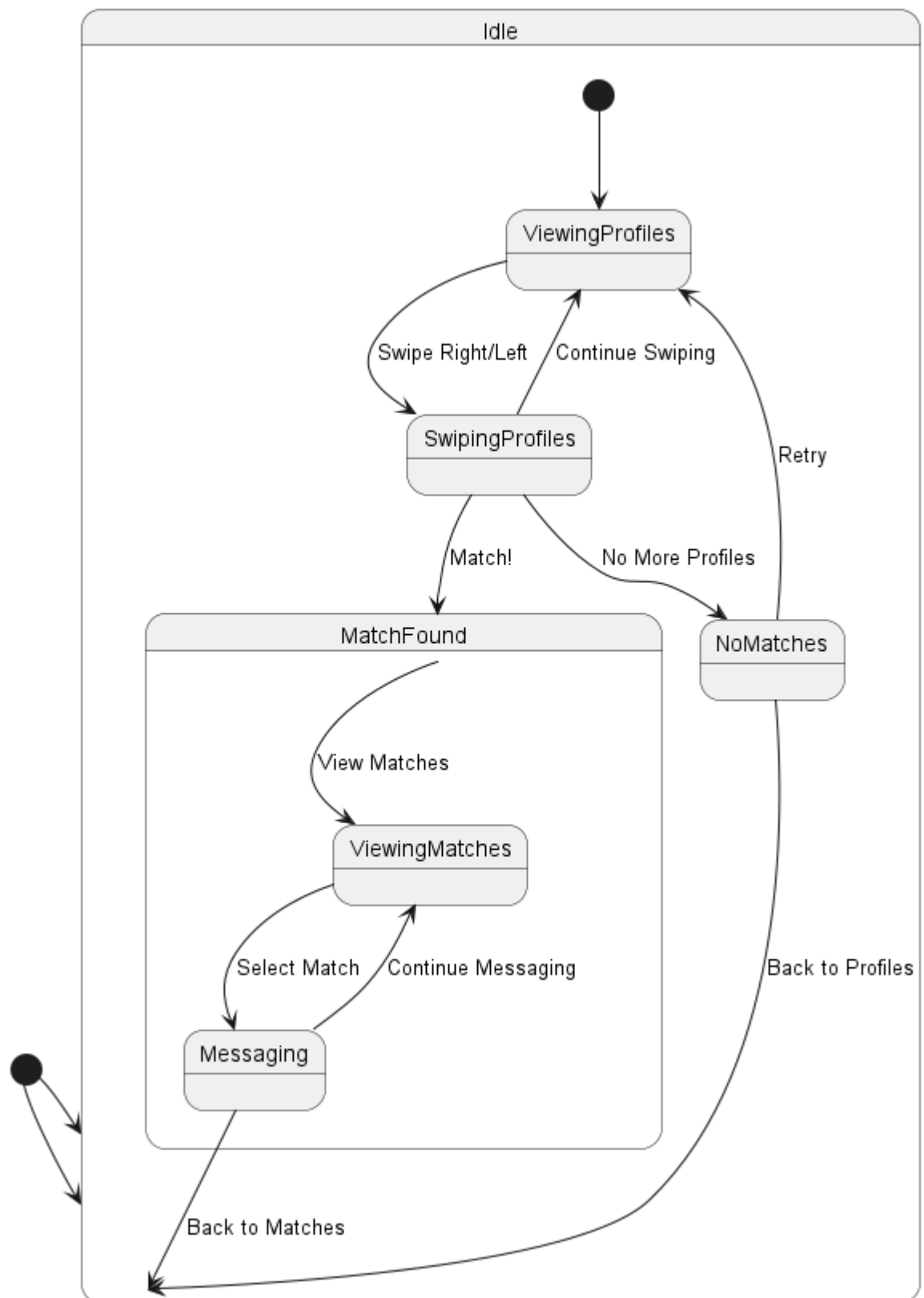
- **User_interaction_ad:**



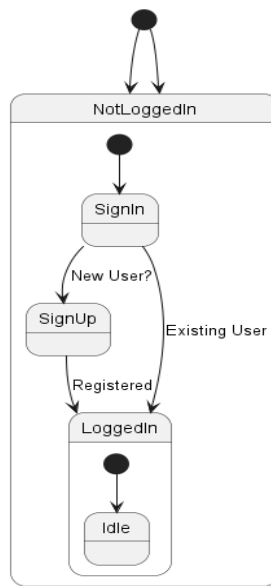
- Sequence Diagram



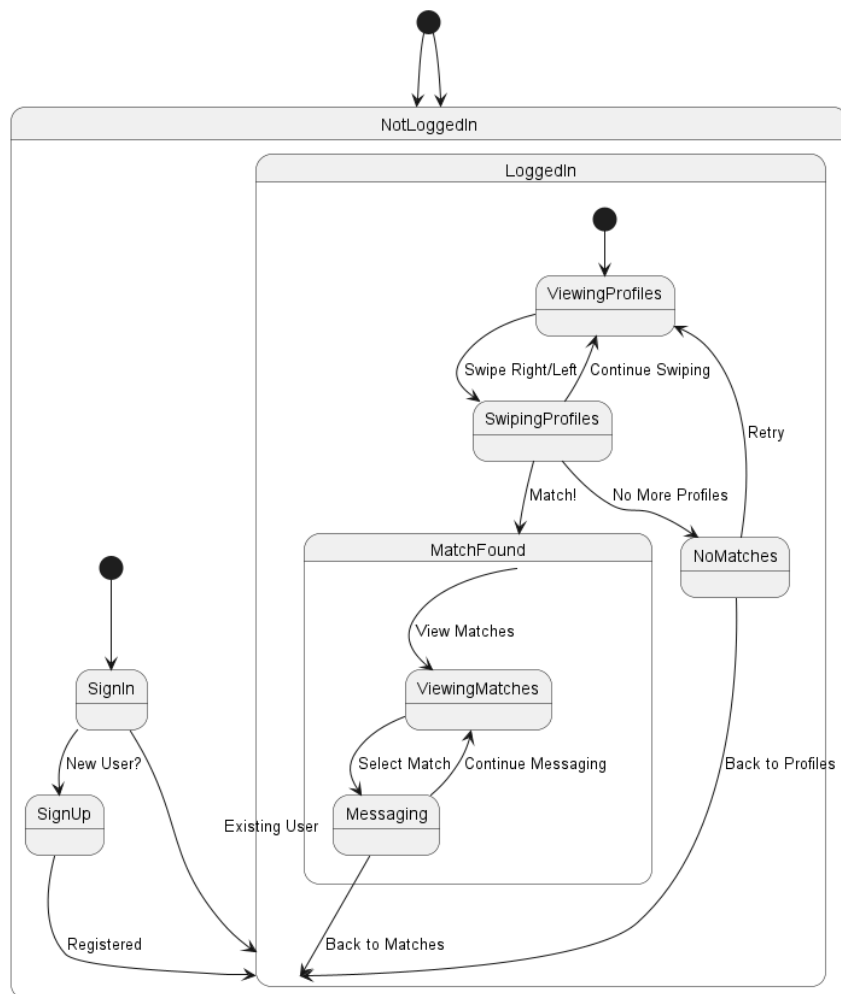
- **State Diagrams**
 - **IDLE_std:**



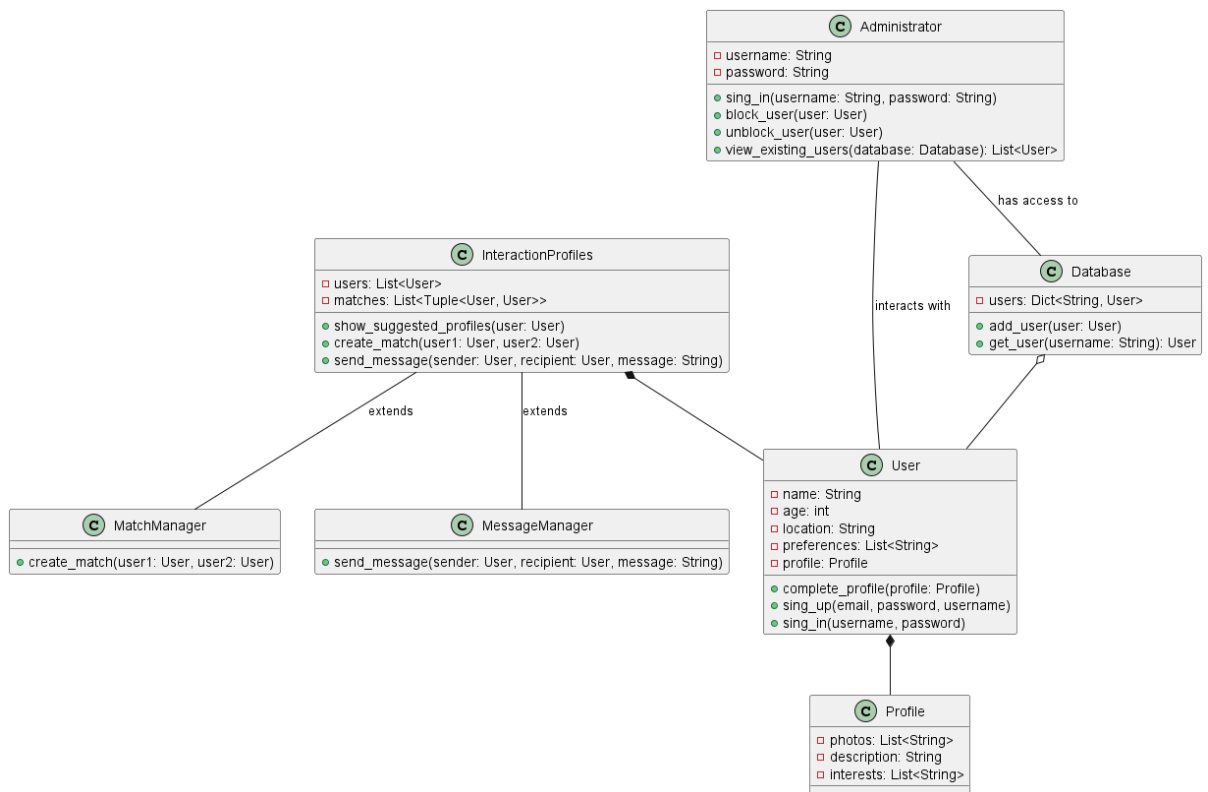
- **Authentication_std:**



- **User_std:**



- **Class Diagram**



CRC Cards

Class: InteractionProfiles	
<ul style="list-style-type: none"> Manage interactions between users, including matching, messaging, and displaying suggested profiles. Handle the creation and management of matches. Facilitate messaging between matched users. 	<ul style="list-style-type: none"> Database User

Class: Profile	
<ul style="list-style-type: none"> Store profile information such as photos, description, and interests. Provide methods for updating and accessing profile information 	

Class: User	
<ul style="list-style-type: none"> Store user information such as name, age, location, and preferences. Manage the user's profile. Handle user authentication (sign in and sign up). Interact with other users, including swiping profiles, handling matches, and sending messages. 	<ul style="list-style-type: none"> User Interface Database InteractionProfiles

Class: DataBase	
<ul style="list-style-type: none"> Store and manage user data, including user profiles and authentication credentials. Provide methods for adding, retrieving, and updating user information. 	<ul style="list-style-type: none"> User InteractioProfiles User Interface

Class: User Interface	
<ul style="list-style-type: none"> Provide the interface for user interaction, including signing in, signing up, viewing profiles, swiping profiles, and messaging. Handle user input and display output to the user. 	<ul style="list-style-type: none"> All classes

Rules of business

The app will contain class-specific information structured as follows:

User: can provide their name, age, location, gender. These data are essential to create a comprehensive and appealing profile for other users.

Interaction Profiles: include information about the user's past interactions on the platform. This can range from the history of matches to exchanged messages and visited profiles. These data help users remember past interactions and make decisions about future connections.

Profile: Each user can create a customized profile that includes a personal description, interests, search preferences, and their current relationship status if desired. This profile provides other users with a fuller picture of who the person behind the photos is.

Data Base: stores all user profiles, as well as interaction records such as matches and messages. Ensuring the security of this data is crucial to protecting users' privacy and maintaining trust in the platform.

User Interface: includes the overall design of the app, as well as specific functionalities such as profile search and filtering, messaging, and notifications. An intuitive and user-friendly interface enhances the user experience and encourages engagement with the platform.

