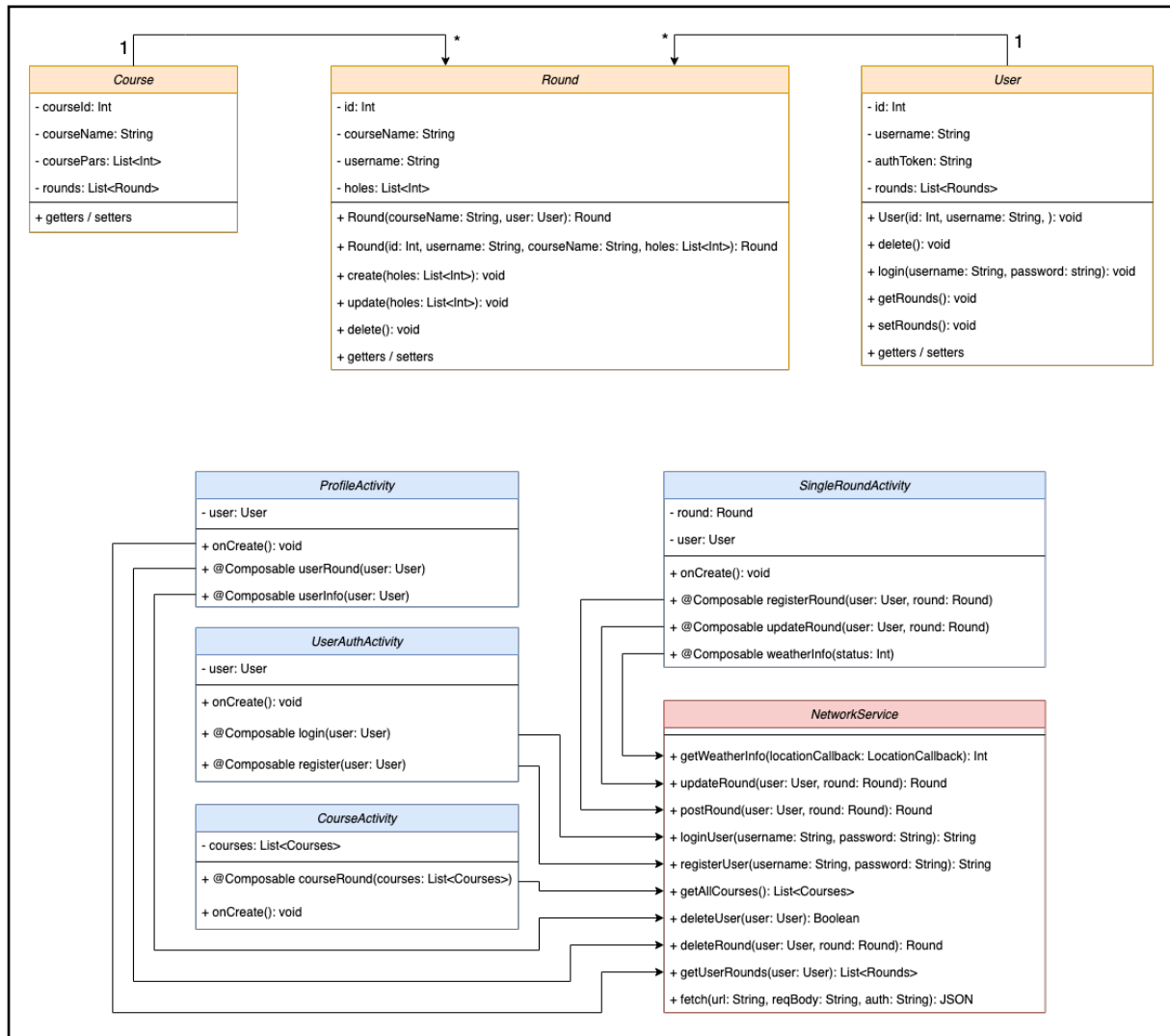


## UML Class Diagram

The class diagram is split into two parts, entities (or models) and controllers and views. The entities are POJO implementations of JSON objects that the golfskor API provides. The controllers and views include UI, business logic that the app uses as well as the network service implementation.



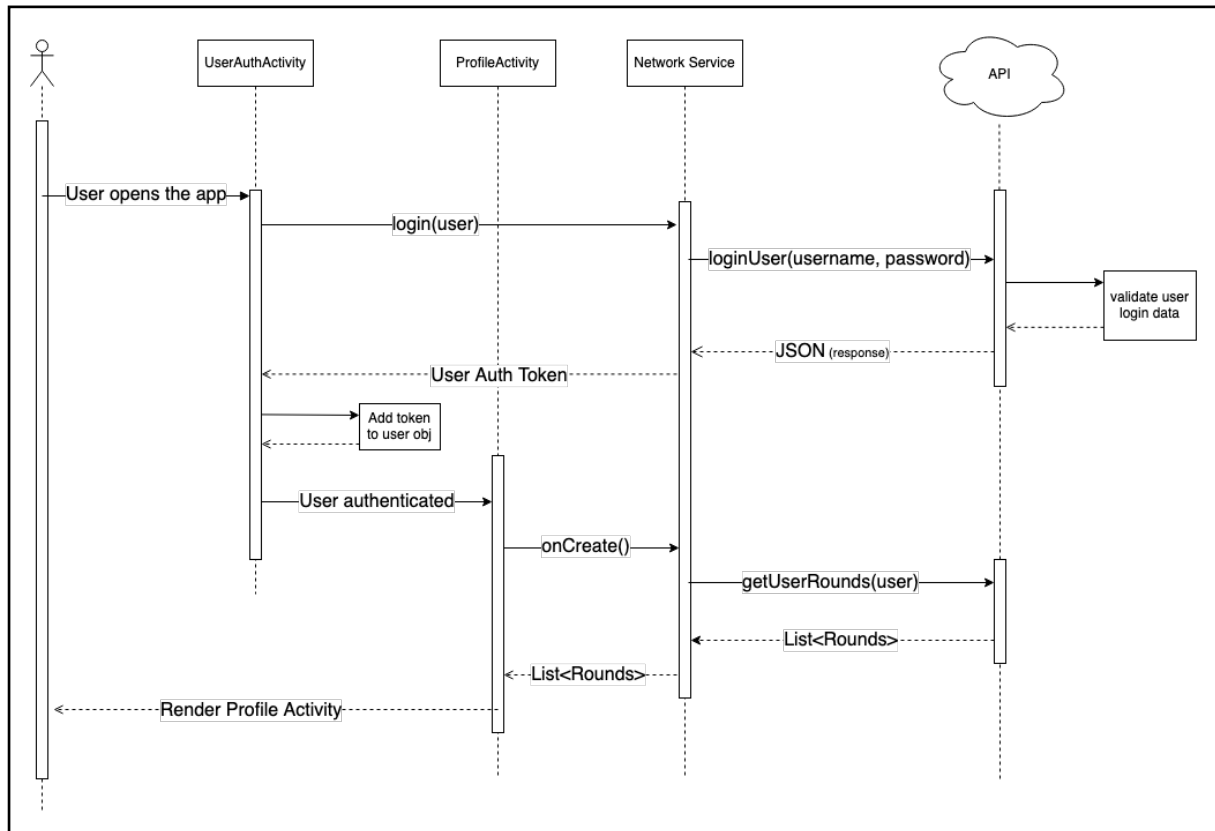
Most of the entity values are set in their respective constructor however both `User` and `Round` have values that need to / can be set via other functions after some process on the client side. This is most obvious in the case of a `User` entity where the `authToken` is not set in the constructor but rather in the `login` function. This is done to simplify the process of creating entities when not talking directly to the server.

For the activity / view layer we are using composables to create our UI blocks. These composable functions are called programmatically and render the UI based on the data that is passed to them.

The network service is implemented as a singleton object that is called from the view model. This is done to simplify the process of making network requests and to make sure that the network service is only ever created once.

## UML Sequence Diagram

This UML sequence diagram shows the flow of the application when a user logs in and fetches a list of rounds. The diagram is split into two parts, the first part shows the process of logging in and the second part shows the process of fetching a list of rounds.



When the user has provided their login info, username and password, the login function is called. This function sends a POST request to the server with the provided credentials and returns a User object with the authToken set. This User object is then used to fetch a list of rounds. When the Profile Activity screen is created, the onCreate function calls the getRounds function which in turn sends a GET request to the server with the authToken and returns a list of Round objects that belong to the user.

## UML State Diagram

The state diagram for the app is rather simple, it requires the user to be logged in to be able to fetch a list of rounds. Once logged in the user can choose between two main screens the Profile screen and the Courses screen. The Profile screen shows the user's information including their previously played games. On the Profile screen users can select old rounds to edit or delete. The Courses screen shows a list of all available courses and the user can select a course to create a new round. There they can also get an overview of all the rounds played on that course.

