U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Bases de Dados

Management of gaming streams

2MIEIC01 - Group 106

March 8, 2020

Gabriel Rocco	up201800172@fe.up.pt
Muriel Pinho	up201700132@fe.up.pt

Description

This project is based on the management of a gaming stream service, similar to *Twitch*.

The system is divided into eight classes, these classes are:

- **User:** it stores all data related to the user, like their id, username, password, profilePicture, birthDate, email and age. The **User** can send and receive a **Message**, he can follow multiple channels and own only one **Channel**, and he can also follow a specific **Game**.

- **Stream:** the stream is the heart of this service, A stream is a live transmission made on a **Channel** owned by a **User** playing a **Game** for other users to watch. A Stream stores a title, startTime, uptime, ageRestriction and viewerCount. A **Stream** has one **Game** being played at a time, but the user can play multiple games over the course of the stream. A stream can also have Tags, describing what the stream is about, with all streams having at least one **Tag**.

- **Message:** it stores the data from messages sent by the **User**, storing the fields content and dateSent.

- **Channel:** a channel is owned by one user and it is where the user can create an **Stream**, the channel stores a biography and followerCount. A **Channel** can have multiple streams but all streams must be streamed on only one channel.

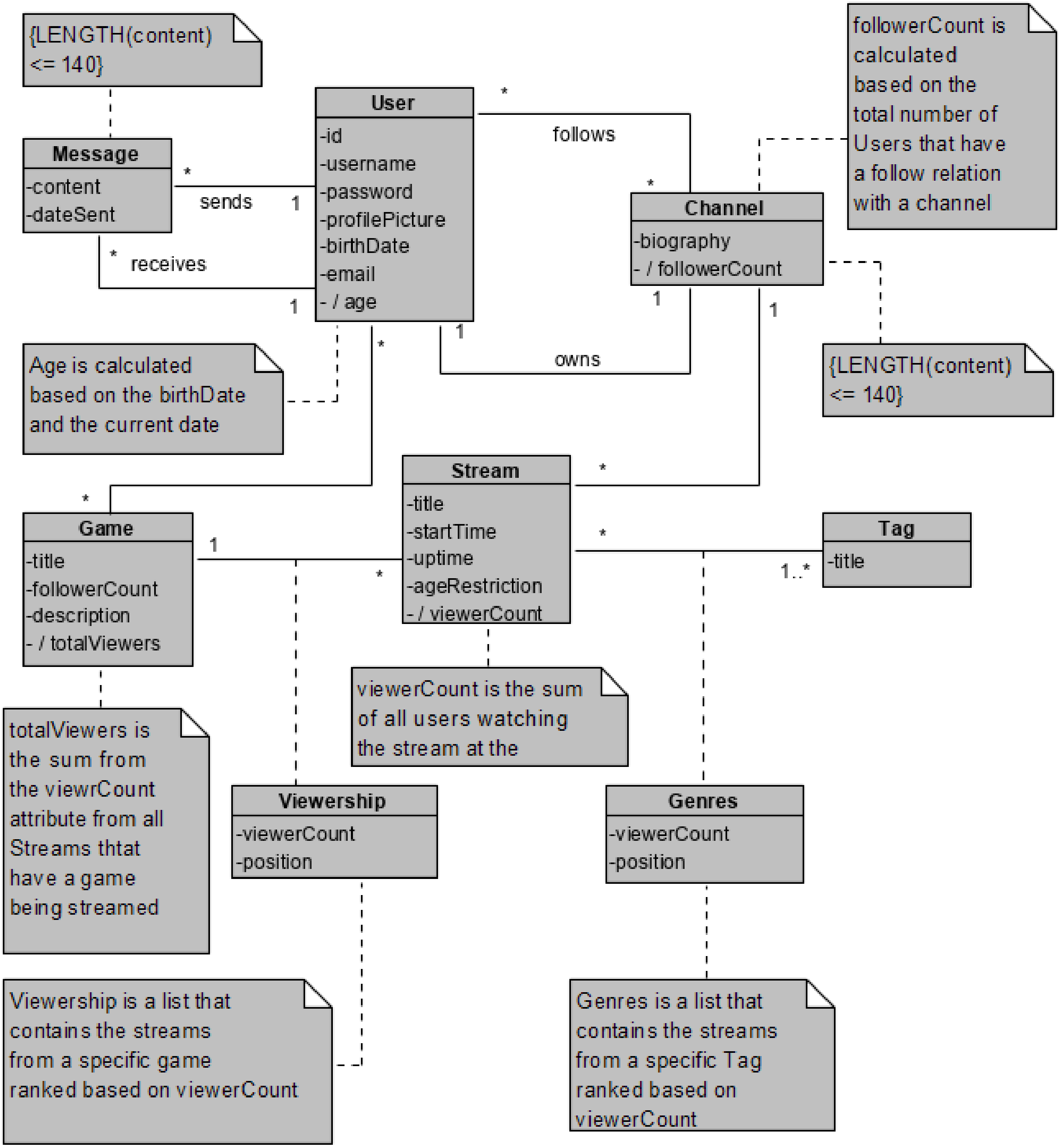
- **Game:** the game class stores all the information related to the games streamed on the platform, a game has a title, followerCount, description and totalViewers.

- **Tag:** a tag is used to describe a **Stream**, it only has a title field and the same tag can be used on multiple streams at the same time.

- **Viewership:** It is used to generate a list that contains the streams from a specific **Game** ranked in order based on the viewerCount from that **Game**.

- **Genres:** it is identical to the **Viewership** class, the only difference is that the ranking is based on a **Tag** instead of a **Game**

Class Diagram - UML


{LENGTH(content)
<= 140}
Message
-content
-dateSent
sends
receives
User
-id
-username
-password
-profilePicture
-birthDate
-email
- / age
follows
owns
Channel
-biography
- / followerCount
followerCount is calculated based on the total number of Users that have a follow relation with a channel
{LENGTH(content)
<= 140}
Age is calculated based on the birthDate and the current date
Stream
-title
-startTime
-uptime
-ageRestriction
- / viewerCount
Game
-title
-followerCount
-description
- / totalViewers
Tag
-title
1..*
viewerCount is the sum of all users watching the stream at the
totalViewers is the sum from the viewrCount attribute from all Streams thtat have a game being streamed
Viewership
-viewerCount
-position
Genres
-viewerCount
-position
Viewership is a list that contains the streams from a specific game ranked based on viewerCount
Genres is a list that contains the streams from a specific Tag ranked based on viewerCount