Chapter 1

Design

1 Main actors

As already mentioned in the introduction we have mainly three major actors: guest user and register user, the latter can be divided between normal user and top critic. In addition there is an admin actor who's main role is to oversee the entire service.

2 Functional requirements

This section describes the functional requirements that need to be provided by the application in regards of the actor:

- Guest (Unregistered) User can:
  - login/register into the service
  - search movies by search bar and other filters
  - view movies, their details and relative reviews
  - view the personal page of the author of a selected *top critic* review
  - view the different halls of fame
- Normal use can:
  - logout from the service
  - search movies by search bar and other filters
  - view movies, their details and relative reviews
  - write a review for a selected film
  - view the personal page of the author of a selected *top critic* review
  - view the different halls of fame
  - follow/unfollow a *top critic* user
  - view the feed of latest reviews from the followed *top critic*
  - view the history of its own reviews
  - modify its own reviews
  - delete its own reviews
  - change its account information

- Top Critics can:
  - logout from the service
  - search movies by search bar and other filters
  - view movies, their details and relative reviews
  - write a top critic review for a selected film
  - view the different halls of fame
  - view the history of its own top critics reviews
  - modify its own top critics reviews
  - delete its own top critics reviews
  - change its account information
  - see the number of its followers
- Admin user can:
  - logout from the service
  - search movies by search bar and other filters
  - view movies, their details and relative reviews
  - view the different halls of fame
  - modify films details
  - add/remove films
  - browse *user* and *top critic*
  - ban *user* and *top critic*
  - register new *top critic*
  - perform analytic on the user population

3 Non functional requirements

In the following section are listed the non functional requirements for the application.

- the system must encrypt users password
- the service must be built with OOP language
- user must have 16 or more years to register into the service
- service must be implemented through a responsive website
- avoid single point of failure in data storage
- high availability, accepting data displayed temporarily in an older version

4 Implementation regarding the CAP theorem

We will now discuss on how we decided to tackle the CAP theorem issue. In our minds the application is a read-heavy one where we expect that the number of read transaction are by far more numerous than the write operation, so we decided that our application would prioritize high availability of and low latency capable of withstanding network partitions. In reference of figure 1.1 it is clear that we moved towards a AP approach in spite of data consistency.

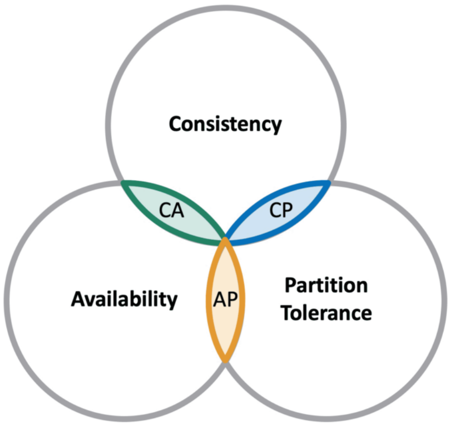
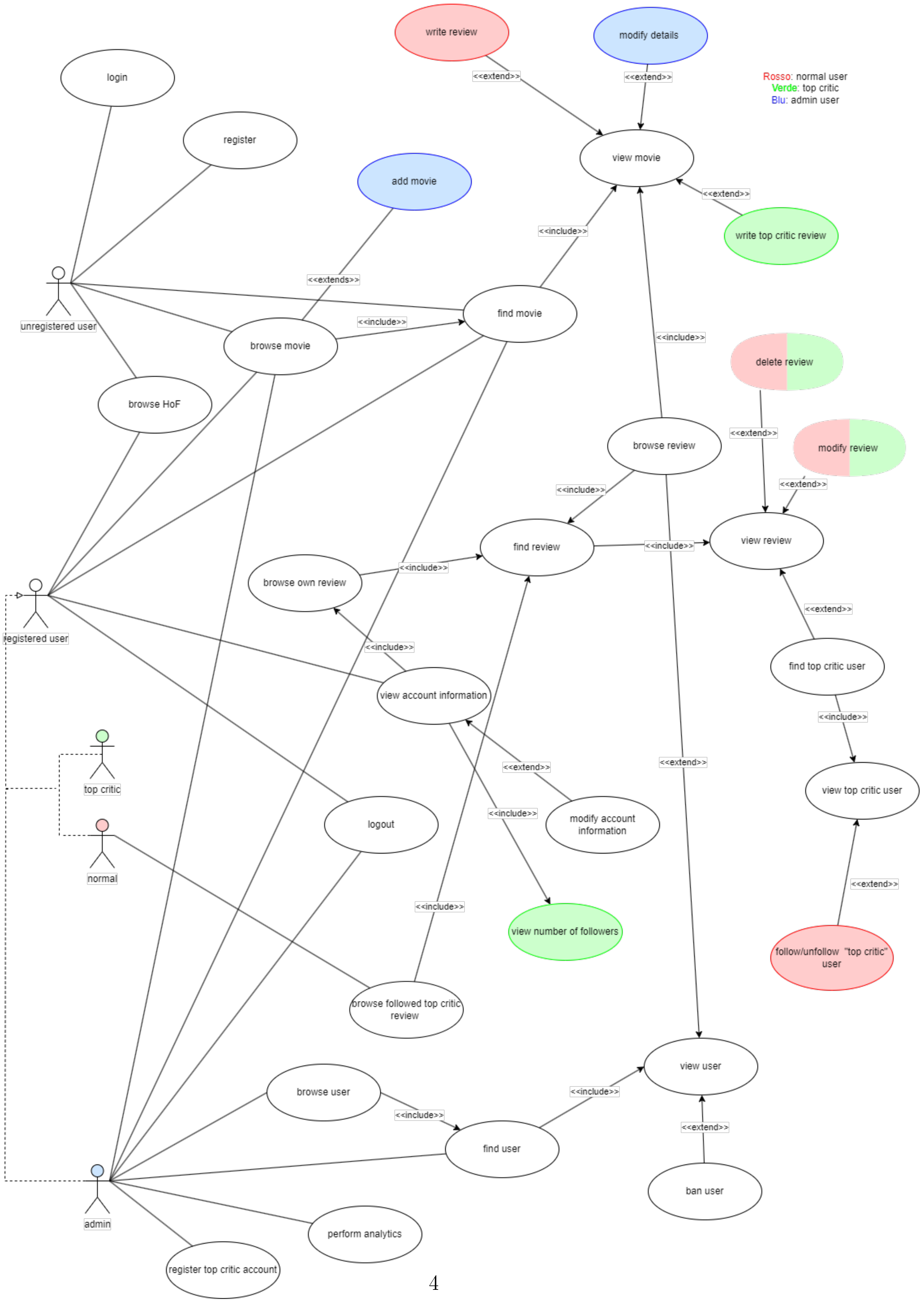


Figure 1.1: CAP theorem diagram

In order to guarantee the requirements of high availability, we decided to accept the cases in witch the data shown to the user could be not updated to the latest version in the database.

5 Use cases



write review
modify details
<<extend>>
<<extend>>
Rosso: normal user
Verde: top critic
Blu: admin user
login
view movie
register
add movie
<<extend>>
write top critic review
<<include>>
<<extends>>
unregistered user
find movie
<<include>>
browse movie
<<include>>
delete review
browse HoF
<<extend>>
modify review
browse review
<<include>>
<<extend>>
find review
view review
<<include>>
browse own review
<<include>>
registered user
<<extend>>
<<include>>
find top critic user
view account information
<<include>>
top critic
<<extend>>
<<extend>>
view top critic user
<<include>>
normal
modify account information
logout
<<extend>>
<<include>>
view number of followers
follow/unfollow "top critic" user
browse followed top critic review
view user
browse user
<<include>>
<<include>>
find user
<<extend>>
admin
ban user
perform analytics
register top critic account

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6 Class analysis

In this section we shall discuss of the design of the various class and how are they related

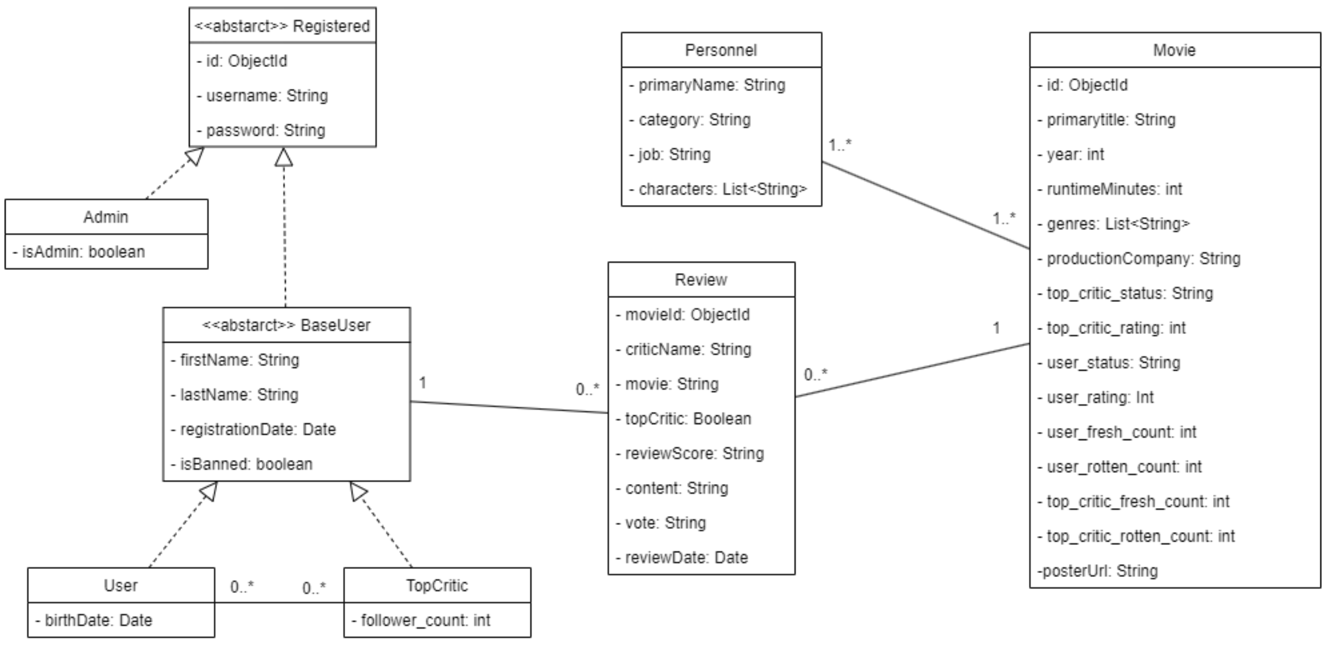


Figure 1.3: Class Diagram

The diagram express the following relationship between the entities.

- a BaseUser can write from zero to many reviews
- a Review can be written by a single BaseUser for a single Movie
- a Movie can have zero to many reviews and can have from 1 to many Personnel working in it
- a Personnel can work in 1 to many Movie