Large-Scale and Multi-Structured Databases

Project Design

My Anime Library

Jacopo Carlon
Ayoub El Ourrak
Nicola Riccardi

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB
Innovation for industry 4.0

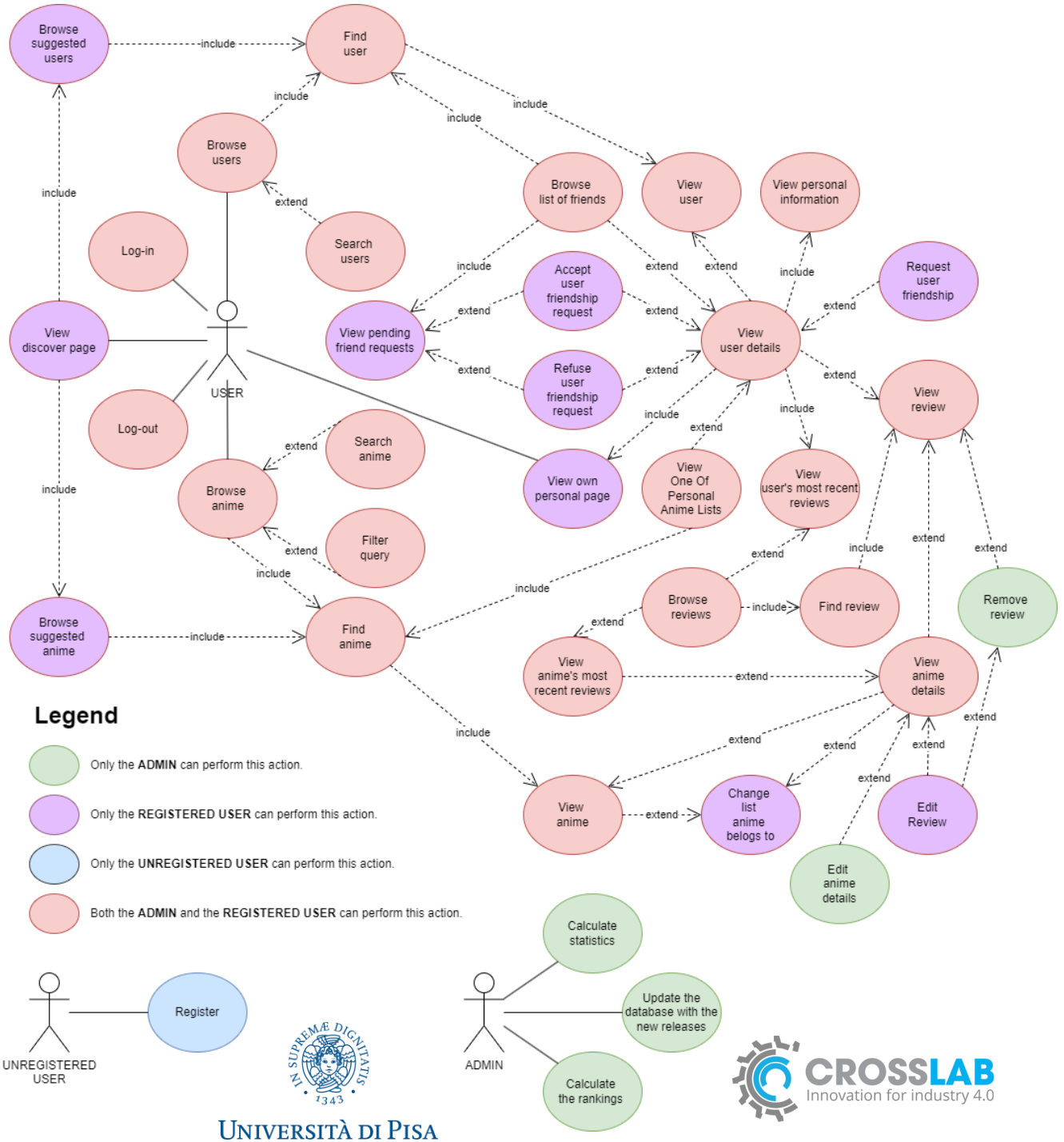
Application Highlights

MyAnimeLibrary is an anime *social-networking* and *social-cataloging* application.
With our website you will be able to:

• **Create Your List**: Our users can create a personalized list from thousands of anime. You will be able to organize and track which titles you've completed, your current progress, what you plan to watch, and much more
• **Join the Community**: Write reviews for the anime you loved the most (or hated), check up on other users' opinions, make friends, and see what they like.

And behind the scenes, our admins keep the library up to date with the newest releases, and the analytics true to YOUR votes.

Actors and main supported functionalities



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

Dataset Description

Source:

- ***https://myanimelist.net/ (anime_info, user_anime_lists, user data, reviews)***
- ***https://sourceforge.net/projects/anime-offline-database.mirror/ (anime_info, images)***
- ***https://1000randomnames.com/ (user data)***

Description:

Dataset contains real anime and users information, as well as list of anime per user.
The name/surname of the users are randomly generated.

Volume:

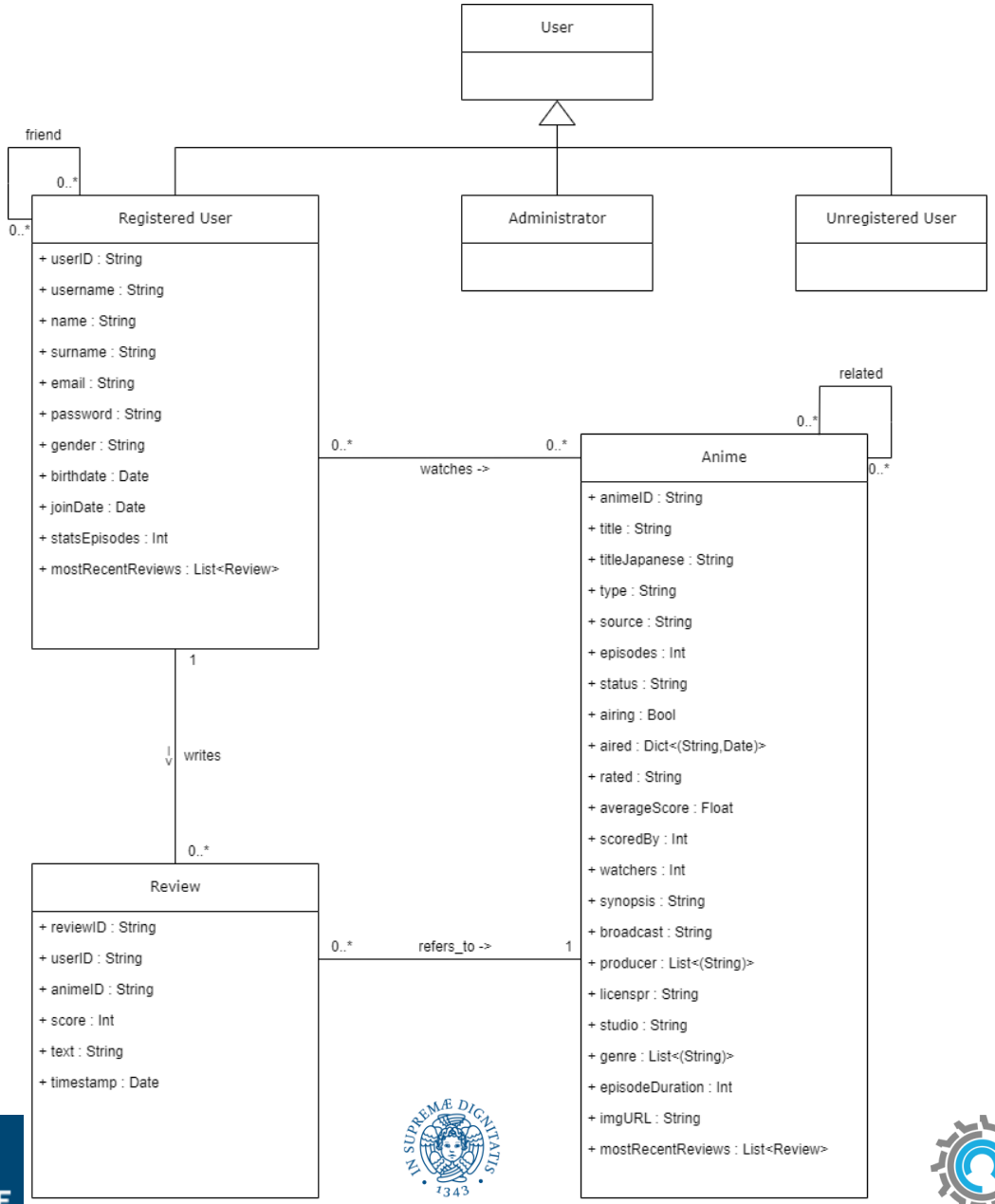
The scraped dataset has three .csv files:

- User_List : **15.75 MB**
- Anime_details : **6.33 MB**
- List of anime for each user : **2.26 GB**
- Reviews : **654,14 MB**
- Anime_info and images : **47.31 MB**
- User_data (125.000 names and surnames): **1.826 KB**

Variety:

Three different sources have been used to build the dataset

Preliminary UML Class Diagram



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB
Innovation for industry 4.0

Non-Functional Requirements

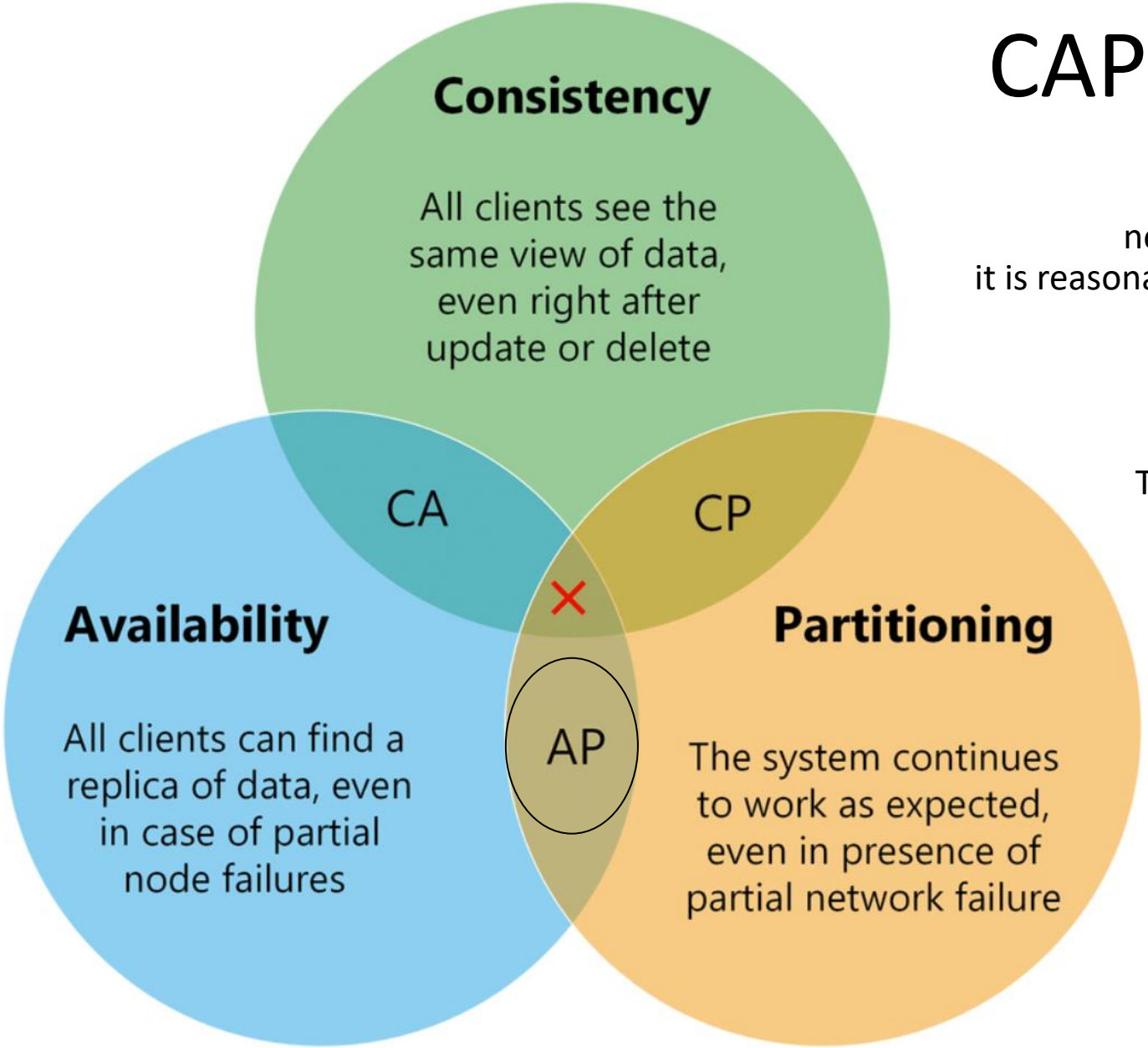
Product Requirements :

- **Usability** : the application must be simple and user friendly
- **High availability** : the displayed data might not be up to date
- **Low latency** : we want a responsive application, therefore accessing the databases must be quick
- **Tolerance** : data loss is tolerated, but we must avoid single point of failure

Organizational Requirement :

- The admin must be able to **quickly correct** erroneous uploads
- If a review is added or removed, the rankings must be eventually updated accordingly
- Encryption of passwords

CAP Theorem



In order to satisfy the non-functional requirements, it is reasonable to sacrifice consistency, in favor of **high-availability** and **partition-tolerance**.

Thus, an **AP solution** is used.

Requirements and Entities handled by Document DB

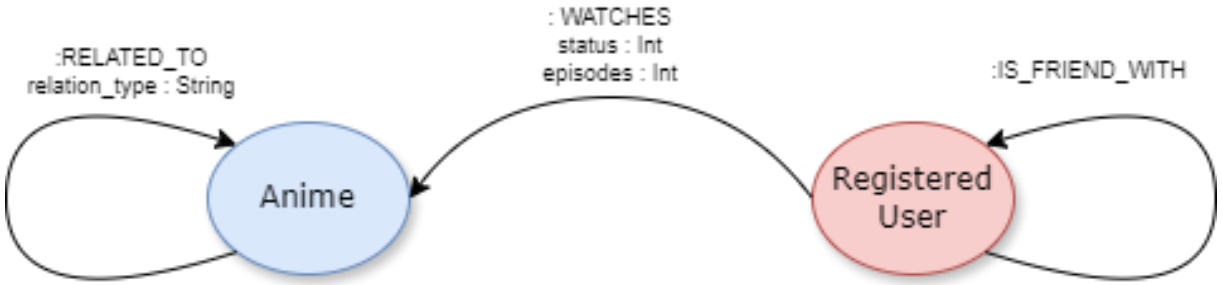
Entities :

- **Anime**
- **User**
- **Review**

Queries :

- Show Anime details
- Show user details/credentials
- Show anime with highest score
- Show user's review per anime
- Show anime number of watchers
- Show anime reviews
- Top anime by genre
- Show most reviewed anime in the last week

Requirements and Entities handled by Graph DB



Entities :

- **Anime**
- **User**

Relationships :

- **RELATED_TO**
- **IS_FRIEND_WITH**
- **WATCHES** (in which list an anime is for a specific user)

QUERIES :

- Show the most watched anime by the most numerous group of friends
- Anime most watched among my friends
- Recommend anime that is related to another one that the user has watched
- Rank friends based on most anime watched
- Recommend new friend based on common tastes
- Show most watched anime franchise
- Show most dropped anime

Software Architecture Preliminary Idea

Programming language :
- Java
- Python

Frameworks :
- Maven
- Java FX
- Spring

DBMS :
- MongoDB
- Neo4j

Web Side :
- HTML
- CSS
- JS

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB
Innovation for industry 4.0