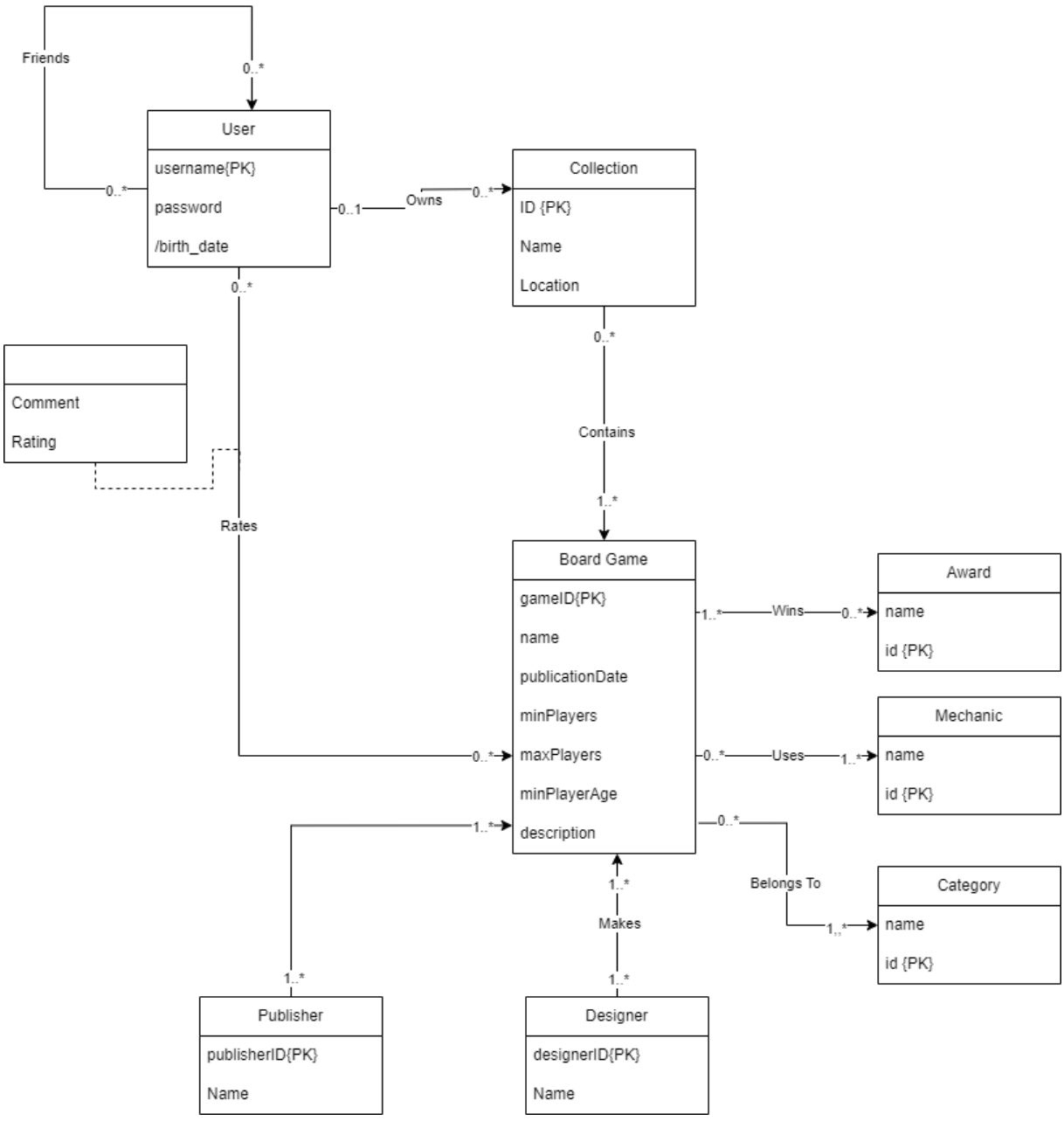
1 Technical Requirements:

- MySQL
  - Can be obtained at `https://dev.mysql.com/downloads/installer/`
  - Recommended version is 8.0.40
  - Install to default directory
- Python3
  - Version 3.12.1 was used for our project
  - Can be obtained at `https://www.python.org/downloads/release/python-3121/`
  - Install to default directory
- py-bgg
  - Recommended version is 0.3.3
  - Install with pip– 'pip install py-bgg'
- PyMySQL
  - Recommended version is 1.1.1
  - Install with pip– 'pip install PyMySQL'

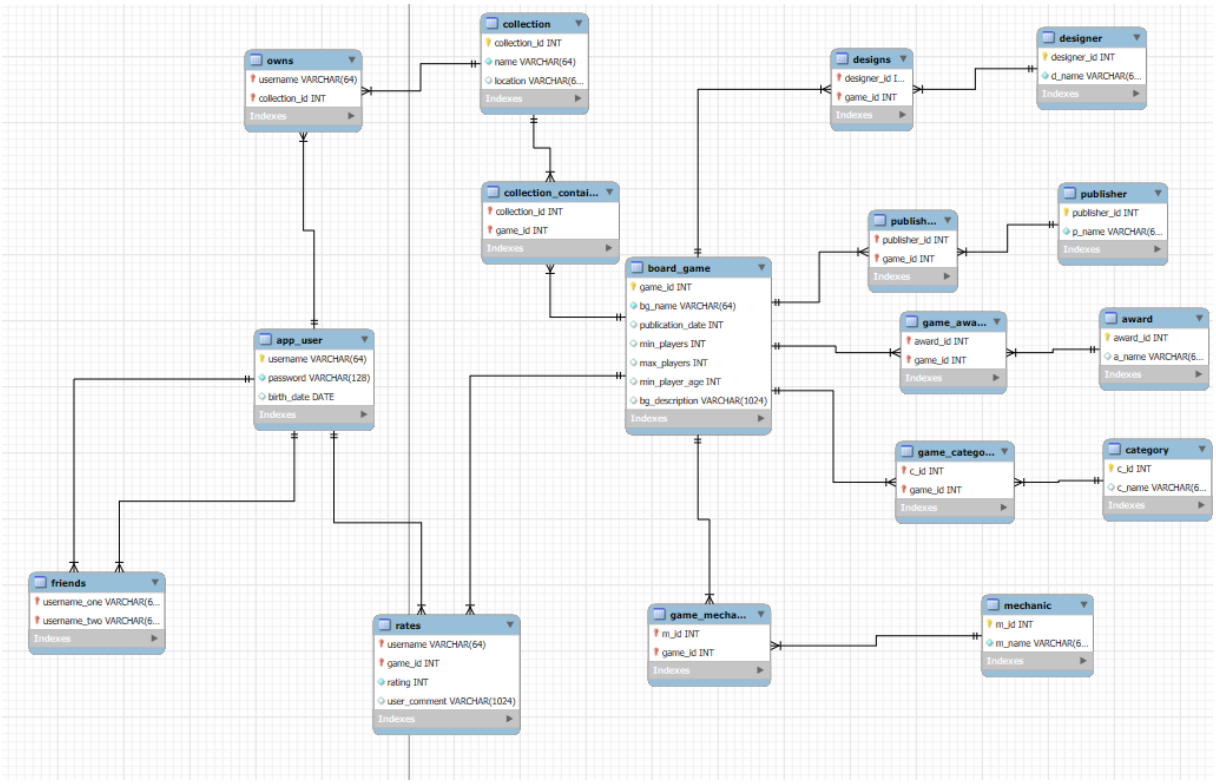
2 Technical Specifications:

This project was written in python3, interfacing with a database written in MySQL

3 UML


Friends
0..*
User
username{PK}
password
/birth_date
0..*
0..1
Owns
0..*
Collection
ID {PK}
Name
Location
0..*
Contains
1..*
Comment
Rating
Rates
0..*
Board Game
gameID{PK}
name
publicationDate
minPlayers
maxPlayers
minPlayerAge
description
1..*
Wins
0..*
Award
name
id {PK}
0..*
Uses
1..*
Mechanic
name
id {PK}
0..*
Belongs To
1..*
Category
name
id {PK}
1..*
Makes
1..*
1..*
Publisher
publisherID{PK}
Name
Designer
designerID{PK}
Name

4 Logical Design



5 User Flow:

- A user launches our application from the command line using python. They would do this with a command similar to 'python3 ./FinalProject/main.py'
- On launching the application, they are prompted to log in or to create a new account. They can select each option by inputting the corresponding numeral.
- Next, they are presented with a series of options. After choosing an option, the associated function runs to completion and the user is again prompted to choose an option.
- Available options are:
  - Adding a friend, which gives the user a list of other users to add to their friends list.
  - Adding a game to a collection. This allows the user to create a new collection of games or to add to a current collection. To add a game, they provide either the title or a substring within the title and are presented with possible games that they might be thinking of. If none of the presented games match the game they have in mind, a large online database is queried to supplement their search. Once they choose a game, it is added to the chosen collection in the sql database.
  - Rating a game. This calls the same game-search from adding a game, then prompts the user for a numeric rating of the game between 1 and 10 and a brief explanation of why they rated it as they did. The rating is inserted into an associated collection in the sql database. If the user tries to rate a game a second time, they will be prompted to edit or delete their rating.
  - Finding a game. This allows the user to search either their friend's collections or all the games in the database for any games which they have not yet rated, sorted by how similar those games are to the user's highly-rated games. This list is then presented to the user.
  - Deleting their account. When this is called, the user must provide their password in order to complete deletion. After deleting their account, the user is logged out.
  - Quitting. Once a user quits, the session is over.

6 Lessons Learned

More than anything else for us, this was a huge lesson in time-management and the importance of controlling the scope of a project. We started with a subject we were extremely excited for, but may have bitten off more than we could reasonably chew by fitting our project to another database. If we could start over, we would have started by reading the project description more clearly rather than starting with an exciting idea. This was not helped by both of us having extremely busy semesters, which left us working with less time than we would have liked.

All that being said, we still had a blast with this project. We have had so much fun finding ways to analyze data in an application that is deeply interesting to us. We've learned to parse huge chunks of data from an API that was not terribly intuitive. While this project might have been easier if we hadn't decided to hook our app into BoardGameGeek, we know that we'll be playing with the program for months to come just to see what else we can learn.

7 Future Work

We don't feel like we're done with this database. We may not be as attached to the runner we built around it, but we would love to throw this database onto a webapp and allow our friends to fill out their own collections.

Both of us host a lot of board game nights, and constantly run into the question of what to play next. We would love to expand the functionality of this database to not just find recommendations for a single person, but instead to search a single collection (say, the games at one person's house) for the game which is most likely to be enjoyed by everyone in attendance.