Manga Database Application

Preyash Patel, Pratik Patil, Hasan Rawjee Fall 2022 CS5200 – Database Management Systems

README: Setting Up and Starting the Application

Prior To Running Program:

Download and unzip the zip file containing the program files (*take note of where you download it*)

Install mySQL Workbench and Python:

- **mySQL Workbench**

  https://www.mysql.com/products/workbench/?p=857

  Import the sql file for the database dump provided in the zip file

  (Server -> Data Import -> Select the dump file)

- **Python**

  https://www.python.org/downloads/

- **Python Libraries**

  Utilize command prompt and pip to install the following libraries (instructions included in links)

  - **PyMySQL**

    https://pypi.org/project/PyMySQL/

  - **Cryptography Package**

    https://pypi.org/project/cryptography/

  - **PrettyTable**

    https://pypi.org/project/prettytable/

Running the Program:

- Locate where you downloaded the program files and right click on the file containing Driver in the name (MangaAppDriver) and select copy as path (this is the file path)
- Open Command Prompt
    - Full screen the window for optimal display results as the data is formatted for tables that are not cut off by an unmaximized window.
- Attempt simply pasting the file path copied earlier and pressing enter (if the program does not run then follow the directions below in Type otherwise skip Type step)
- Type:

    py (right click/paste/manually type the file path here) and press enter to start

    [Note py could be python or something else if you renamed it on your computer]

- Program should be started at this point assuming directions were followed

    1. The program will ask for a user name and password for the connection to the mySQL server. If invalid credentials are entered, then the program must be restarted.

    2. Once the connection has been established, the application will be running and you may login, create an account, etc. and explore the application. If you would like to see the entire pathing in the program visually, reference the user flow diagram in this report)

    *(**Note CRUD Testing**: Sample user and admin login information has been included below or you may create a new user account to interact with; however, this will be more work and take more time to test as you will start with an empty reading list, so we suggest you use the sample user information to test the application)*

    User: Login = zoro Password = Zoro@1234 | Admin: Login = luffy Password = luffy$123

Project Description

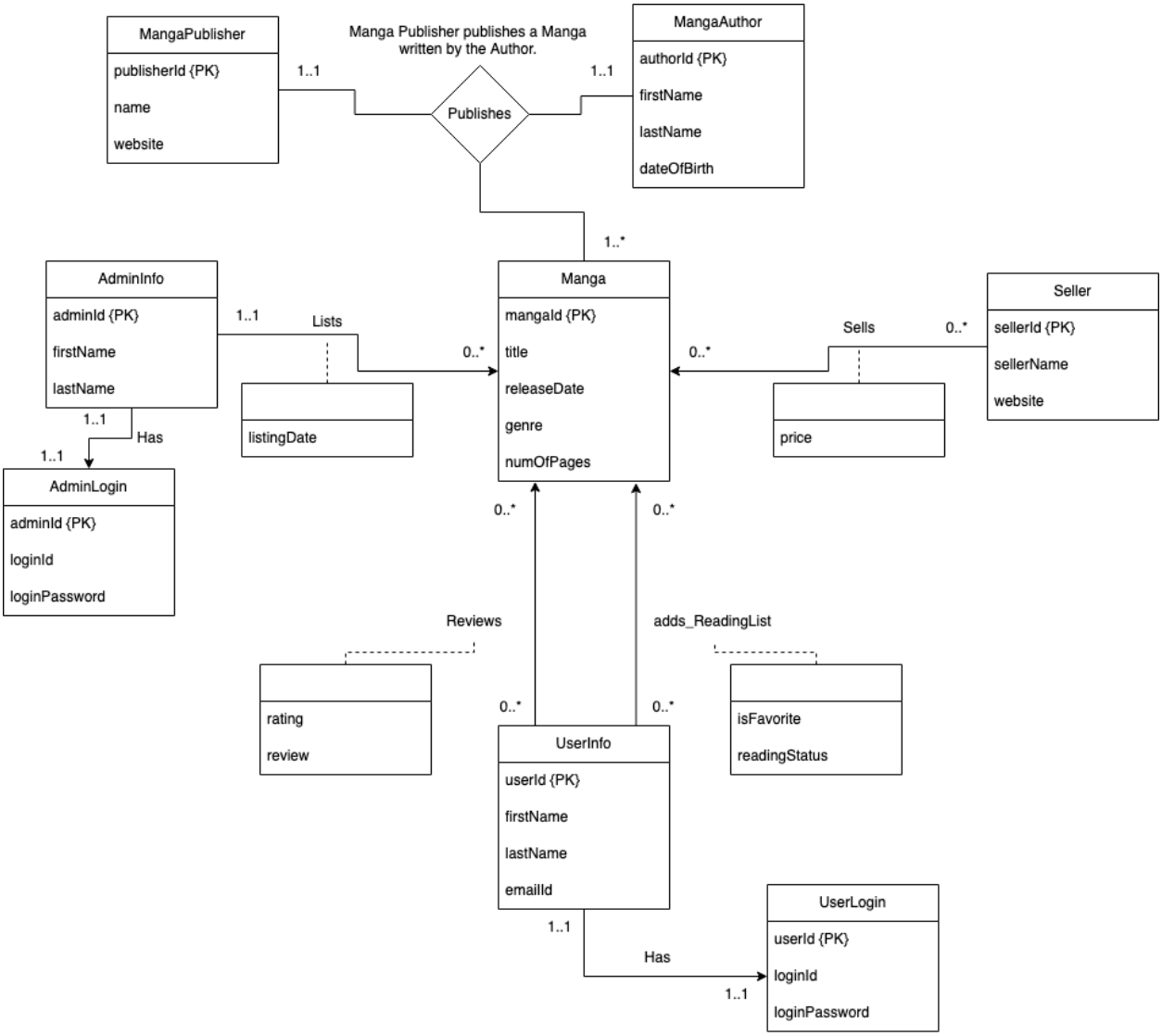
Domain Interest

We enjoy reading manga in our free time and believed that this would be an interesting and enjoyable topic to pursue as a project application. Though manga is not as popular here as it is in Japan, it has become more popular over the last few years with the increasing interest in anime due to companies like Netflix, Amazon, Crunchyroll, etc. releasing and hosting popular anime content. Our goal is to create a successful application that can be used to track manga we are reading in addition to functionality to rate & review them in an organized manner that we can easily reference.

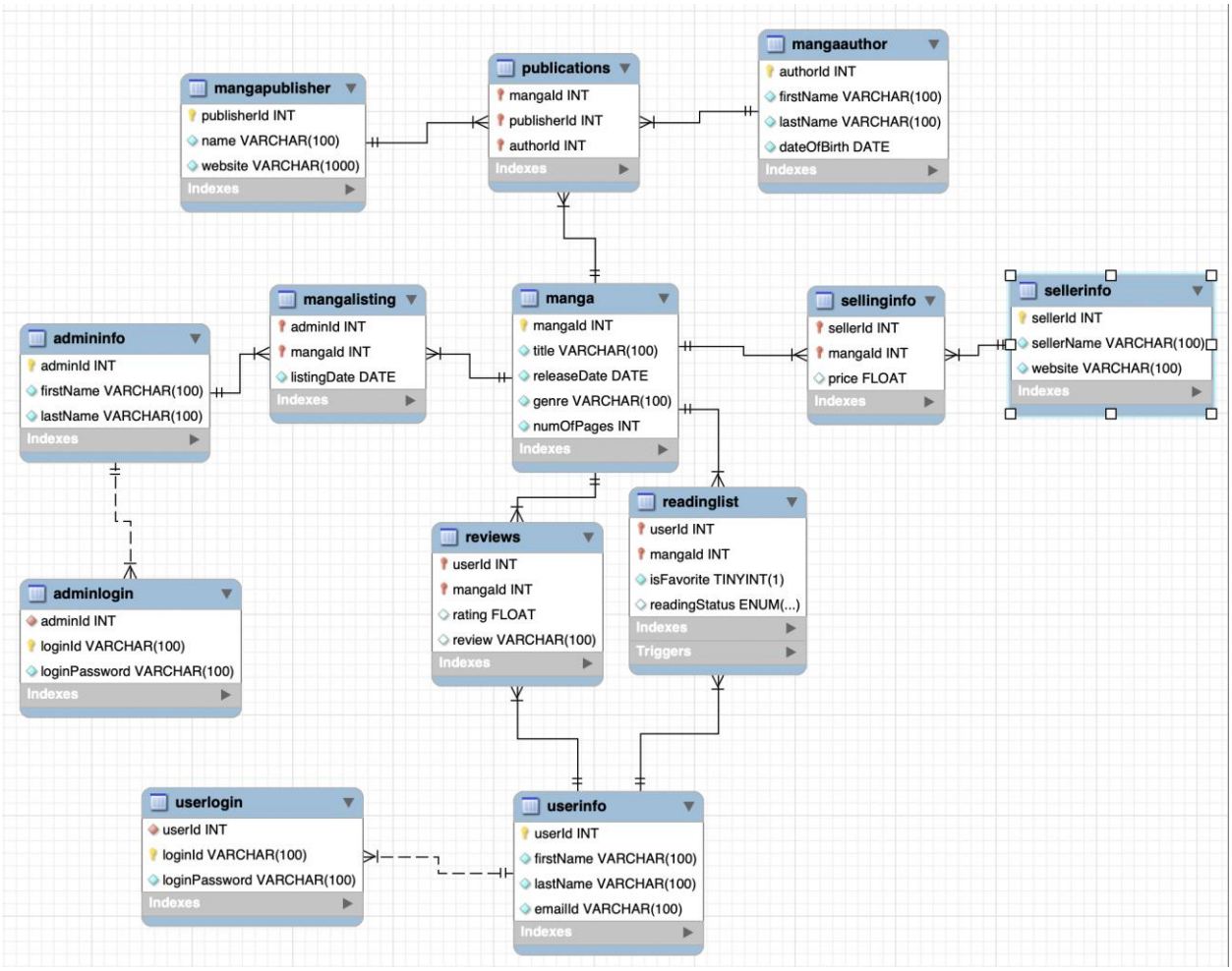
Technical Specifications

This manga application utilizes SQL and mySQL workbench to create and manage the relational database named mangaproj. The application was designed to be a command line interface application coded in python. Several python libraries are used to aid in the implementation of the application. For instance, PyMySQL is used to connect python to the mySQL database containing the application's data. In addition, the cryptography package for python is used to aid in the login functionality of the program. Another library that is vital to this program is the PrettyTable library for python. A function was created in python to properly parse query results from the database and use the PrettyTable library to display the data to the user in an organized and well formatted manner.

Entity Relationship Diagram (UML Format)



Logical Design



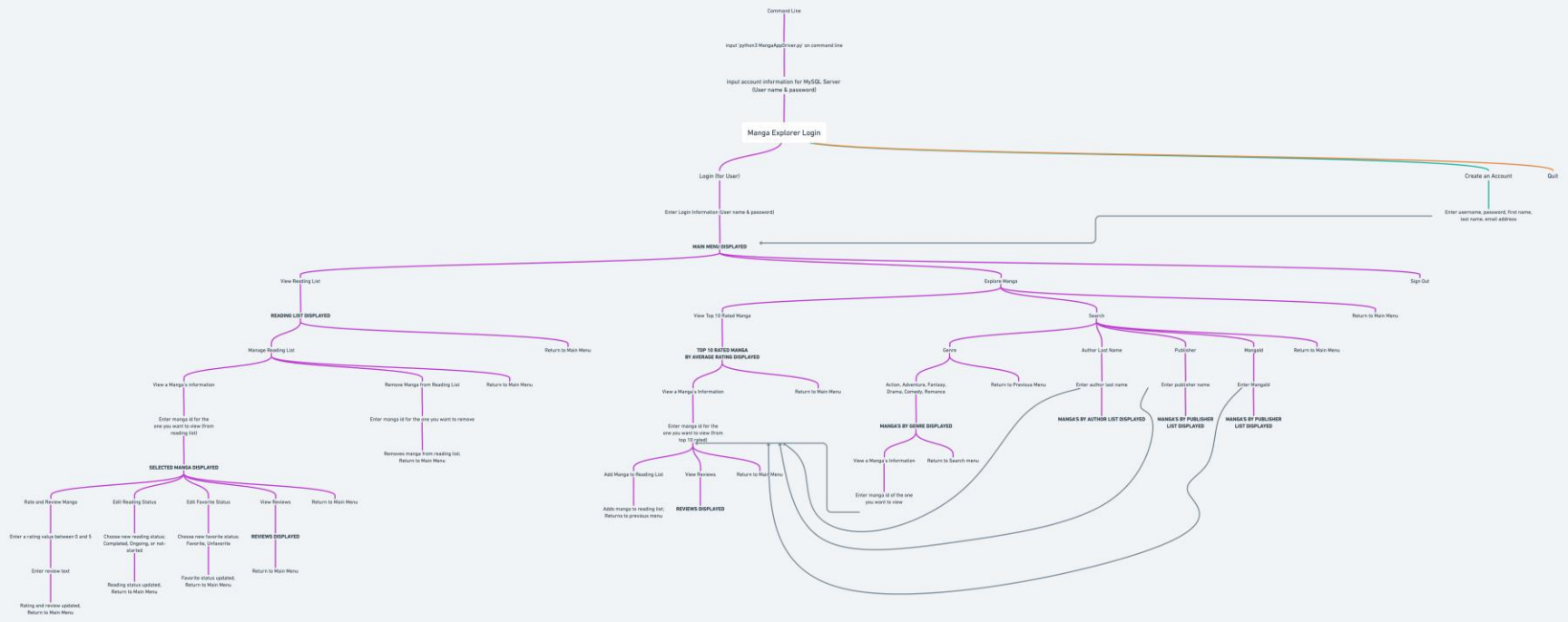
Database Design Description

1. Manga is the main table of this database which contains list of mangas, with information about manga and mangaId. This table contains mangaId which is auto-incremented integer, representing unique identification for each entry. Apart from this the table contains manga title, release date, genre, and number of pages.
2. The MangaPublisher and MangaAuthor tables represent the publisher and author details respectively. MangaPublisher has publisherId which is unique for each publisher which is auto-incremented integer, it also has name and website of the publisher. MangaAuthor has first name, last name, date of bith and mangaId as unique identification for table and it is auto-incremented.
3. Publisher, author and manga connections are stored in table Publications, which is auto-updated by the procedure as you add the manga with required details.
4. Admin information is stored in adminInfo table which has adminId which is a unique identifier which is auto-incremented, also it has first name and last name of the admin.
5. AdminInfo table is connected with adminLogin table, which contains loginId and loginpassword for respective admins. This has adminId as primary key as well as foreign key. Entry in this table is connected only after entry in the adminInfo table is added. And the new entry is mapped in the procedure.
6. MangaListing is the table which represents in relationship attribute in ERD diagram, this table has adminId and mangaId as composite primary key and automatically stores the date of listing for current manga.
7. SellerInfo table contains name and website of the seller, and for unique identification it uses sellerId which is auto-incremented.
8. SellingInfo table represents relationship attributes from ERD diagram between tables manga and seller. This table has sellerId and mangaId as composite key, and price as attribute. This table has to be created separate as one seller can sell multiple manga and one manga can be sold by multiple sellers.
9. User information is stored in userInfo table which has userId which is a unique identifier which is auto-incremented, also it has first name, last name and emailId of the user.
10. UserInfo table is connected with userLogin table, which contains loginId and loginpassword for respective users. This has userId as primary key as well as foreign key. Entry in this table is connected only after entry in the userInfo table is added. And the new entry is mapped in the procedure.
11. Reviews table represents relationship attribute of one of the relationship between user and manga table, it contains attributes rating and review, rating is out of 5 and reviews can be null.
12. Reading list represents another relationship attribute between user and manga table, it contains attributes isFavorite and readingStatus, which have Boolean and Enum values respectively. ReadingStatus can be ‘not-started’, ‘ongoing’, ‘completed’.

`

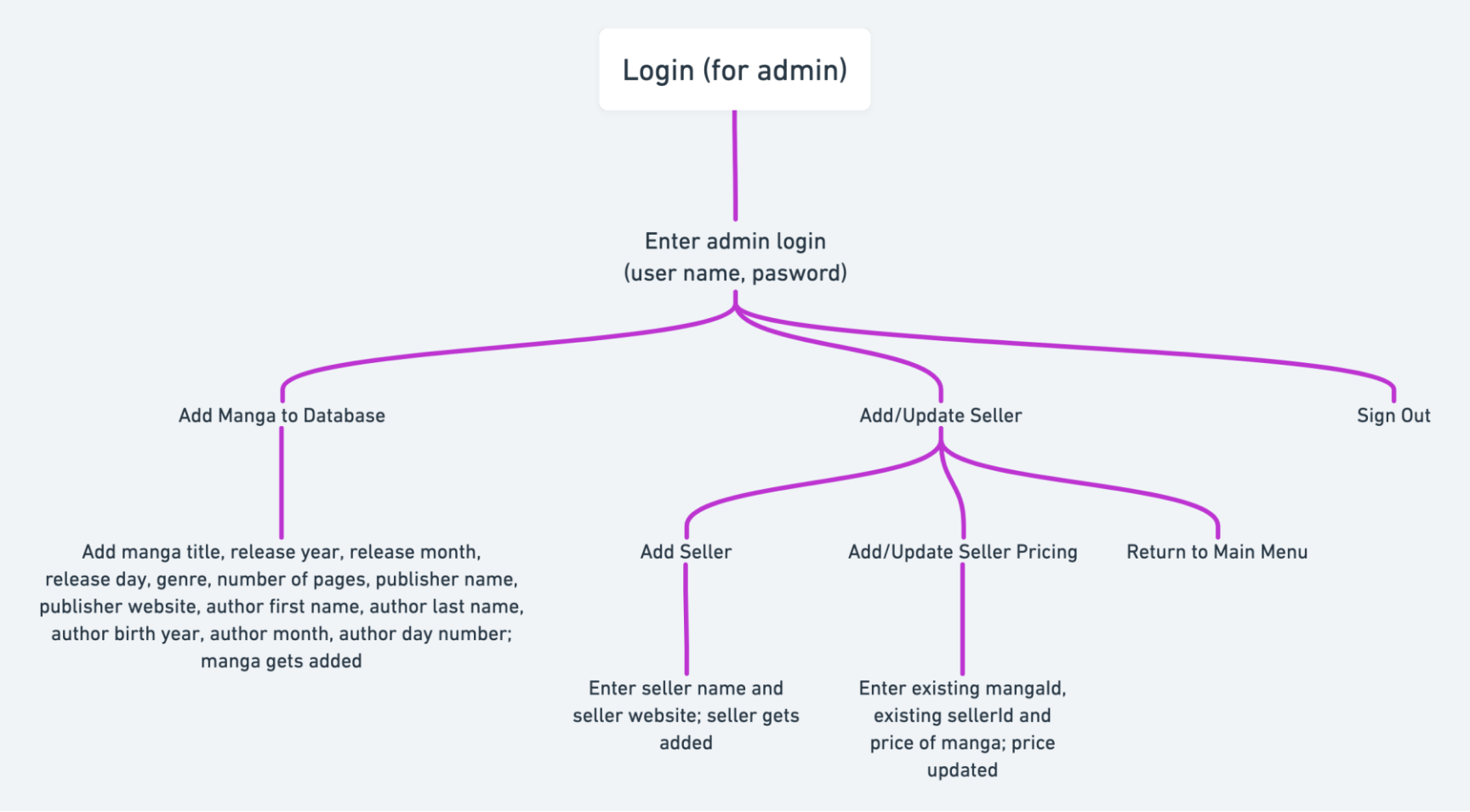
User Flow Diagram

(***Note*** *– Full Resolution Diagram has been included in the submission separately as the diagram is very large and cannot fit on this page to scale)*



Admin Flow Diagram

(***Note*** *– Full Resolution Diagram has been included in the submission separately if needed)*



`

Lessons Learned

This application was more complicated than we had originally anticipated. However, due to this fact, we were able to gain a lot more experience with SQL, python, and successfully communicating between a database and a python program. For instance, our database includes more than 10 tables and as a result, many of our SQL queries required multiple joins from many different tables to successfully obtain the data from the database. This allowed us to gain more experience writing such queries, many of which were more complex than anything we had previously done. Similarly, we were able to gain more experience writing procedures and triggers as they were vital to performing application functions and queries. Additionally, we were able to learn about and practice using various python libraries that aid in connecting an application to a database and modifying it accordingly. Time management was a slight issue for this project as it was more complex than we anticipated and this was our first time creating an application that connected to a database, requiring more time to complete. However, for the most part, we implemented all the original functionality that we intended. In fact, we added some functionality such as an extra filter for users to search, adding sellers and updating pricing in the database (for the admin).

Future Work

We plan to use our application to keep track of manga that we are reading and build it up over time to include more data and functionality. We would like to implement functionality to extract manga data to build our database out with thousands of different manga in the future. This application is fairly easy to use with clear menu options; however, it would be more user friendly as a website, especially when considering scaling the application. In fact, we had decided that if we were able to finish a command line interface application early, then we would attempt to extend the project into a website. However, we were not able to do that in time for the project deadline. All of us are applying for internships next semester and plan to extend this project over winter break so that we can further our knowledge and build a sufficiently large project that we can include on our resumes.