Milestone 2

1. Motivation
   We found a strong database crawled from the website “Book-Crossing” and thus want to build a web app to “visualize” the database so that readers can explore it. The web app will gather most essential functionalities that facilitate readers as they explore, search for books of interest and keep track of the reading list as well as provide lists of facts/ ranking/ recommendation based on the existing book database for them to browse and explore.

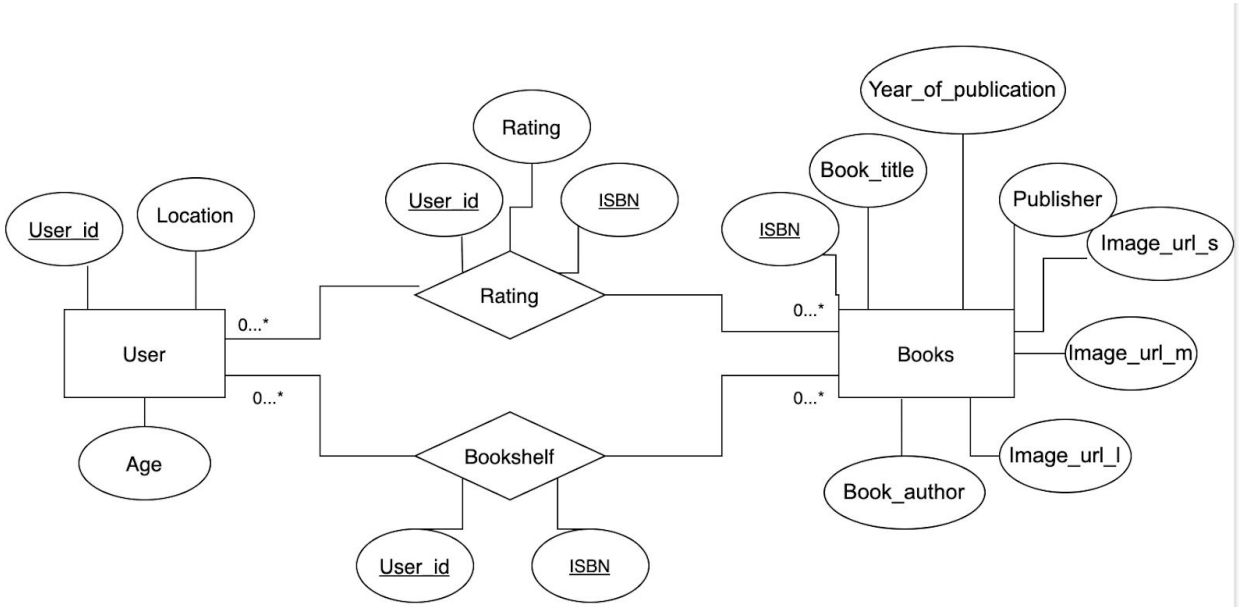
2. List of features will definitely implement
   A) A list of top 10 books of highest rating over all years
   B) A list of highest rated book per decade
   C) Search a book by title
   D) Search a book by author, returns all books of the author
   E) Search a book by year range
   F) Search a book by publisher, returns all books of the publisher
   G) Detailed information of book: when you click on the cover of the book, a detailed description of the book will pop up, which includes title, author, publisher, rating, and other books written by the same author. (? need additional database)
   H) New users can sign up, and his or her information will be added into the User, including an assigned user name, location, age, email, and password.
   I) Users can add a book to the bookshelf. Bookshelf stores all the books marked by the user.
   J) Users can also be able to rate a book after adding it to the Bookshelf. The rating by the user will be added to the rating table in the database.
   K) Recommendation based on the user’s bookshelf. Recommendation based on the user’s bookshelf. Based on the current user’s bookshelf, return books from other similar users(whose bookshelf have large overlap with current user) that are not in the current user’s bookshelf.

3. List of features might implement
   A) Recommendation based on similarity of content in bookshelves between the user and all other users. I.e. For the current user A, find the user/ a set of user B that has the most overlap in bookshelf with A, returns to A the other books that B likes but not in A’s bookshelf.

B) In part 2(G), which is the detailed description, we may also add the description of the book, however, we need to search for an additional database about description of books.

4. List of page
   A) Main page (feature A., B., and G.):
      This page is a fact page, which will show top 10 rating books over all years and highest rating books per decade. Clicking on the book cover will navigate to the info page of the book, which would describe in part (B).
   B) Subpage of book information(feature I):
      A subpage can be triggered by a click on any book cover. It lists all the information about the book including the total rate calculated from the rating table, and has a UI element to add the book to the bookshelf. Below book info, there's a list of other books written by the same author. The rating of the book will be shown on this page if the user rated it before.
   C) Search page (feature C. -> F. and G.):
      The user can search the book by title/author/year range/publisher on this page, which will return a list of books. By clicking, users can navigate to part(B)'s subpage.
   D) Bookshelf page (feature H. and G.):
      This is the page for showing the list of books that the user had already added to their bookshelves. Again, the user can navigate to part(B)'s subpage by clicking.
   E) Recommendation page (feature J. and G.):
      On this page, users can find the books that they may like. Again, a click on the book cover navigates to part(B)'s subpage.
   F) Login page:
      Users could sign in or sign up on this page.

5. Relational schema as an ER diagram



6. SQL DDL for creating the database

```
create  table user(
User_name varchar(20)
Location   varchar(50),
Age    int,
email varchar(50),
Password varchar(20)
PRIMARY KEY (User_name)
);

create table books(
ISBN   int,
Book_title   varchar(20),
Book_author   varchar(20),
Year_of_publication   int,
Publisher   varchar(20),
Image_url_s   varchar(100),
Image_url_m   varchar(100),
image _url_l   varchar(100),
PRIMARY KEY (ISBN)
);

create table rating(
User_name   int,
ISBN   varchar(13) NOT NULL,
```

```
Book_rating   int,
PRIMARY KEY (User_name, ISBN)
FOREIGN KEY (User_name) REFERENCES user(User_name),
FOREIGN KEY (ISBN) REFERENCES books(ISBN),
);

create table bookshelf(
User_name   int,
ISBN   varchar(13),
PRIMARY KEY (User_name, ISBN)
FOREIGN KEY (User_name) REFERENCES user(User_name),
FOREIGN KEY (ISBN) REFERENCES books(ISBN),
);
```

7. Explanation of how you will clean and pre-process the data
   For the data cleaning, we will use pandas as described in the tutorial to remove rows with missing value for “keys” indicated in the ER diagram. This means we will remove rows with missing user_name or ISBN in three dataframes (i.e. book_df, rating_df, and user_df). For other columns with missing value, for example, the age and location in the user table, we will ignore them since the user might be unwilling to provide those information, and the image urls as well. However, if we are missing book_rating in the rating_df table, we will remove the row since the row is meaningless if there is no rating.

   For the entity resolution, since user_name and ISBN are the int type, they are always consistent in three tables. Thus we do not need to resolve the inconsistency. Also, we will not remove unpaired entities. Since the user table provides additional information we could use (e.g. location, age), we will not remove user_name that only appears in the user table but not in the rating table.

   Because we want to show our text-based categorical variables as the information of books to our users (e.g. publisher), we will not replace any categorical variables with indicators.

   Since we have existed users and new users.
   Then we will check that the keys we selected in the ER diagram are unique. For the single-column index, using .unique() to make sure user_name is unique in the user table and ISBN is unique in the book table. For multi-column index, group rating dataframe by user_name and ISBN, then calling (grouped_rating.size()==1).all(). In this case, we can also make sure that the key {user_name, ISBN} is unique in the rating dataframe.

Finally we will export our new dataframe as CSV for later use in MySQL Workbench.

8. List of technologies you will use

   Html, CSS, Vue for front-end

   Mysql for database

   Python(Pandas to preprocess data)

   Java (with Spring boot framework) to connect front-end and back-end

9. Description of what each group member will be responsible for

   ① Setting up front end environment and build up frames/components of pages

   ② Setting up backend environment, cleaning and populating data

   ③ Main Page (feature A., B., and G.)

   ④ Search page (feature C. -> F. and G.)

   ⑤ Bookshelf page (feature H. and G.)

   ⑥ Recommendation page (feature J. and G.)

   ⑦ Subpage of Book Info (feature G.)

   ⑧ Login page

   Zhi Zheng: ① ③ ⑧

   Kejing Wang: ① ④ ⑧

   Zhifan Xu: ② ⑤ ⑦

   Chen Fan: ② ⑥ ⑦