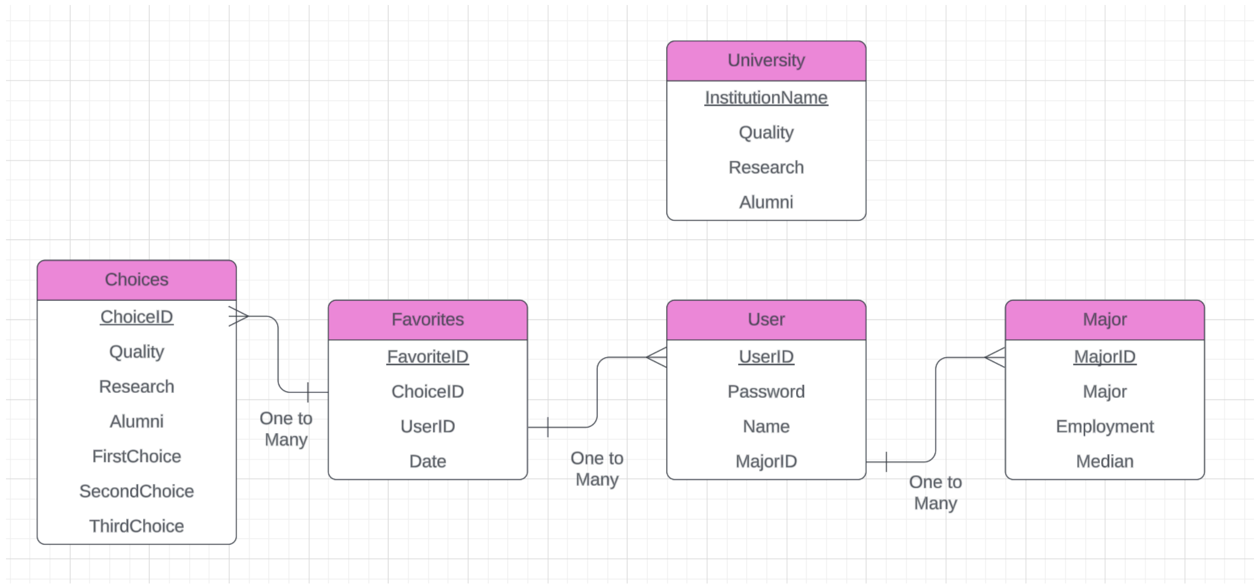
UML Diagram:



Assumptions:

- Each user can have multiple favorites
- Each favorite corresponds to one user
- Each favorite corresponds to one choice
- Each choice can have multiple favorites
- Each user has one major
- Each major can have multiple users

Rationale for each Entity:

User:

User is modeled as its own entity rather than an attribute of another entity to hold relevant information regarding the user of the application. These attributes are the Password of the User, the User's name, and the User's major, which are not attributes of another entity.

Major:

Major is modeled as its own entity because it stores relevant information regarding specific majors. Multiple users can have the same major, making it more efficient to model it as its own entity, and relate it through the MajorID attribute.

Favorites:

Favorites is modeled as its own entity because it stores multiple favorites that correspond to a single student. This matches the assumption that a student can store multiple favorites. Each favorite stores the criteria that the student would like to rank by.

Choices:
Choices is modeled as its own entity to store precomputed rankings of the top 3 universities based on specific criteria like quality, research, and alumni. Since these rankings are based on boolean attributes that are unlikely to change frequently, precomputing them allows for faster retrieval when users access the data. The role of the Choices entity is to offer users a way to select which ranking criteria they are interested in (quality-focused or research-focused rankings). By storing these precomputed results as separate choices, the system efficiently handles user preferences without needing to dynamically rank universities each time a user makes a selection.

University:
University is modeled as its own entity because it stores the information regarding each university, which will be used to create the base ranking that the user sees without making any favorites. These could not be stored as an attribute of any of the above entities.

Cardinality:

- User to Choices is a many to many relationship because each user can have multiple choices, while each choice can have multiple users
- User to Favorites is a one to many relationship because each user can have multiple favorites, but each favorite can only have one user
- Favorites to Choices is a many to one relationship because each favorite corresponds to exactly one choice, but each choice can have multiple favorites.
- User to Major is a many to one relationship because each user has one major, but each major can have multiple users

Normalizing Database:
For our dataset, our schema adheres to the BCNF normal form because for every non-trivial functional dependency X→Y, X is the superkey, which means that the key determines the attribute in every dependency that we have sketched out. All functional dependencies are preserved and the schema can be reconstructed without any loss of information. Below each relation's normalization is described according to BCNF.

user = USERID → {PASSWORD, NAME, MAJORID}
University = INSTITUITIONNAME →{ QUALITY, RESEARCH, ALUMNI }
Major = MAJORID → {MAJOR, EMPLOYMENT, MEDIAN}
favorites = FAVORITEID →{ CHOICEID, USERID, DATE}
Choices = CHOICEID → {QUALITY, RESEARCH, ALUMNI, FIRST CHOICE, SECOND CHOICE, THIRD CHOICE}

Relational Schema:

```
User (
        UserID: INT [PK],
        Password: VARCHAR(255),
        Name: VARCHAR(255),
        MajorID: VARCHAR(255) [FK to Major]
)

University (
        InstitutionName: VARCHAR(255) [PK],
        Quality: INT,
        Research: INT,
        Alumni: INT
)

Major (
        MajorId: INT [PK],
        Major: VARCHAR(255),
        Employment: INT,
        Median: REAL
)

Favorites (
        FavoriteID: INT [PK],
        ChoiceID: INT [FK to choices],
        UserId: INT [FK to user],
        Date: INT
)

Choices (
        ChoiceID: INT [PK],
        Quality: BOOL, Research: BOOL,
        Alumni: BOOL,
        FirstChoice: VARCHAR(255),
        SecondChoice: VARCHAR(255),
        ThirdChoice: VARCHAR(255)
)
```