William Pan

PART A: The requirements document as a PDF.

Problem: Dining hall food on college campuses usually suck. But what if people who ate the food throughout the day rated the food so future people eating later in the day could decide if they want to go spend money/food swipes on the dining hall food.

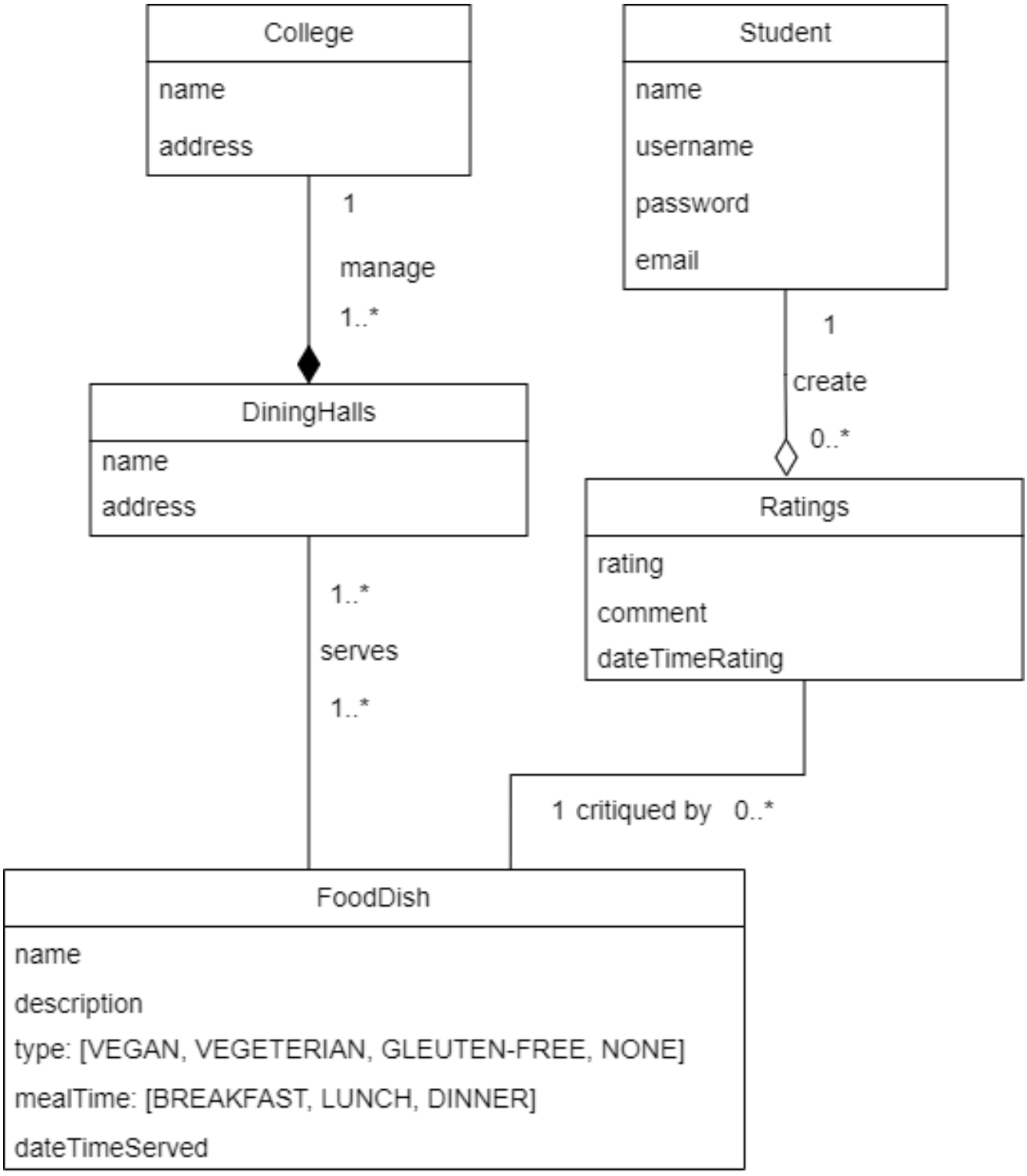
Objectives: Build a database that stores a list of colleges and dining halls and food that each college has. Students can make their own account and rate food from dining hall from 1-5 stars. Over time, people can know what dishes are the best/worst and what times/days dishes are usually served.

Nouns: college, city, address, dining hall, food, students, date,
Verbs: rating, commenting, serving, attend

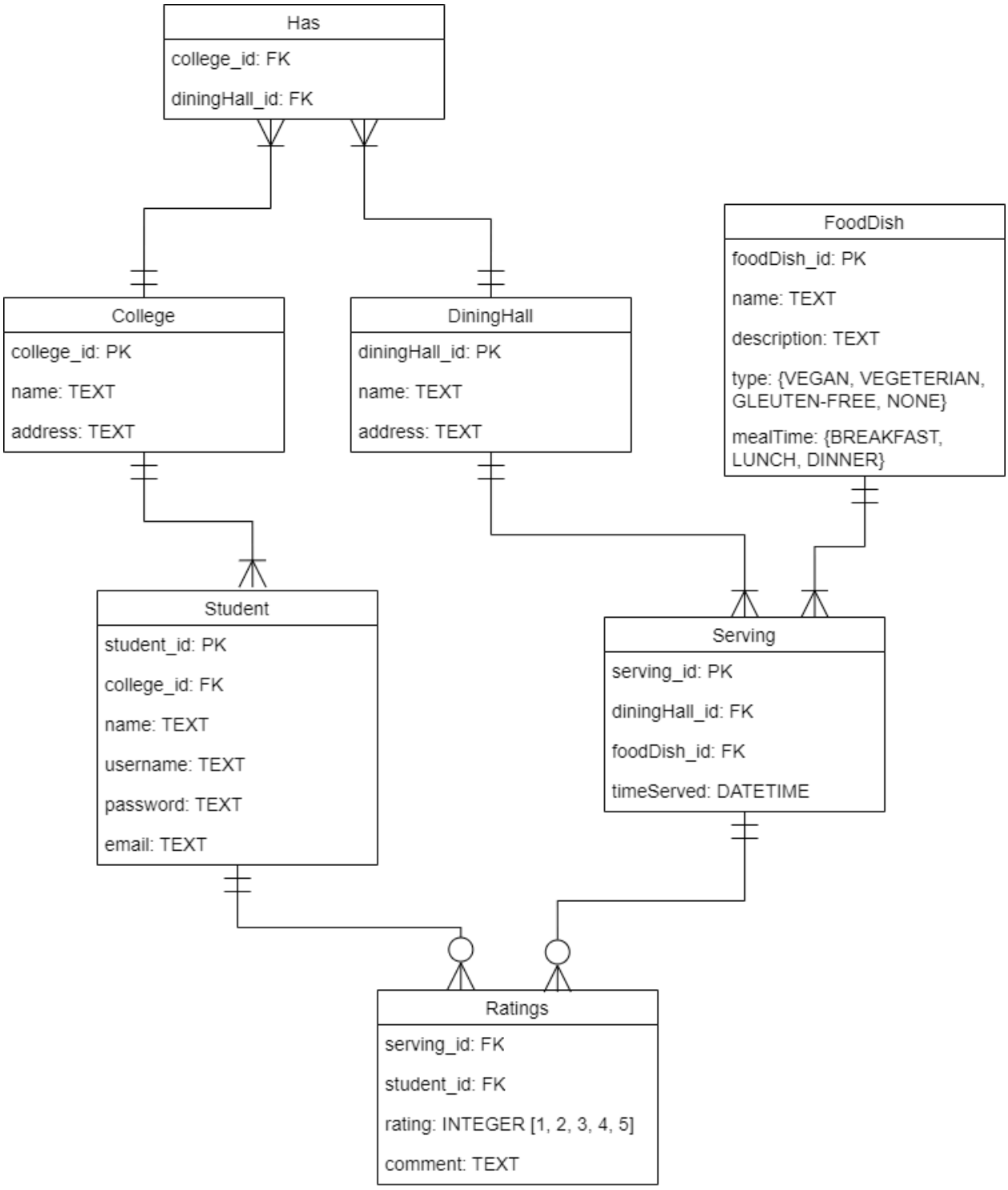
Rules:

1. One college can have many dining halls
2. One dining hall can only be in one college
3. One college can only have one address (one campus)
4. One college can have multiple students
5. One student can only attend one college
6. One dining hall can have multiple food dishes
7. One food dish can be served at multiple different dining halls
8. One food dish can have multiple servings (a food dish can be served more than once)
9. Users can comment and rate multiple foods but can only rate one serving once
10. Food ratings must be an integer from 1 to 5
11. Food rating comments are optional (can be null)
12. Food meal times can be classified as breakfast, lunch and/or dinner
13. Food types can be classified as only vegan, vegetarian, gluten-free, or diet restriction

PART B: UML Class Diagram as an embedded JPG/PNG.



PART C: ERD as an embedded JPG/PNG and URL to LucidChart diagram.



ERD LINK (draw.io):
https://drive.google.com/file/d/1Hw0eYextoUM1kIlGpAZFbr9JkXkRj1yB/view?usp=sharing

PART D: Definition of relational schema with proof that it is in BCNF.

College(
 college_id (PK): INTEGER,
 name: TEXT,
 address: TEXT
)

DiningHall(
 diningHall_id (PK): INTEGER,
 college_id (FK): INTEGER,
 name: TEXT,
 address: TEXT
)

Student(
 student_id (PK): INTEGER,
 college_id (FK): INTEGER,
 name: TEXT,
 username: TEXT,
 password: TEXT,
 email: TEXT
)

FoodDish(
 foodDish_id (PK): INTEGER,
 name: TEXT,
 description: TEXT,
 type: TEXT, -- {VEGAN, VEGETARIAN, GLUTEN-FREE, NONE}
 meal_time: TEXT -- {BREAKFAST, LUNCH, DINNER}
)

Serving(
 serving_id (PK): INTEGER,
 diningHall_id (FK): INTEGER,
 foodDish_id (FK): INTEGER,
 time_served: DATETIME
)

Ratings(
rating_id (PK): INTEGER,
serving_id (FK): INTEGER,
student_id (FK): INTEGER,
rating: INTEGER, -- {1, 2, 3, 4, 5}
comment: TEXT

College Table

Functional Dependency: college_id -> name, address

BCNF Explanation: The College table is in BCNF because college_id is the primary key and every attribute in the table (name, address) is functionally dependent on it.

DiningHall Table

Functional Dependency: diningHall_id -> name, address

BCNF Explanation: The College table is in BCNF because diningHall_id is the primary key and every attribute in the table (name, address) is functionally dependent on it.

Has Table

Functional Dependency: college_id, diningHall_id -> NONE

NONE because there are no other attributes to depend on since the 2 foreign keys are the primary composite key

BCNF Explanation: The Has table is in BCNF because (college_id, diningHall_id) is a superkey with no additional attributes.

Student Table

Functional Dependency: student_id -> college_id, name, username, password, email

BCNF Explanation: The Student table is in BCNF because student_id is the primary key and every attribute in the table (name, username, password, email) is functionally dependent on it.

FoodDish Table

Functional Dependency: foodDish_id -> name, description, type, meal_time

BCNF Explanation: The FoodDish table is in BCNF because foodDish_id is the primary key and every attribute in the table (name, description, type, meal_type) is functionally dependent on it.

Serving Table

Functional Dependency: serving_id -> diningHall_id, foodDish_id, time_served

BCNF Explanation: The Serving table is in BCNF because serving_id is the primary key and every attribute in the table (diningHall_id, foodDish_id, time_served) is functionally dependent on it.

Ratings Table

Functional Dependency: rating_id -> serving_id, student_id, rating, comment

BCNF Explanation: The Ratings table is in BCNF because rating_id is the primary key and every attribute in the table (serving_id, student_id, rating, comment) is functionally dependent on it.