CS 5600
Triet Ngo
Dr. Guerra Gomez
3 Nov 2023

Requirements and Conceptual Document
CATabase: A Cat Adoption Database Management System

I. Requirements

a. Problem Description

There are approximately 50-70 million feral cats in the United States, with some figures estimating at least 100 million without a home. Cats are natural predators and thus pose a serious threat to the local wildlife while also being at risk of being attacked by other cats, aggressive animals, or seriously injured by fast-moving vehicles. And yet, there are only about 2 million cats being adopted each year in the US. As such, a comprehensive cat adoption database will help people more easily connect with owners and shelters to adopt and reduce the number of vulnerable cats out in the elements.

b. Summarized Nouns and Verbs

i. Nouns

1. Cats
   (1) cat_id: unsigned integer
   (2) cat_name: string
   (3) cat_breed: {“Siamese”, “Bengal”, “American Shorthair”, “Russian Blue”, …}
   (4) cat_birthday: mm/dd/yy
   (5) cat_weight_lb: float
      (a) Note: Weight in pounds
   (6) cat_owned_since: mm/dd/yy
      (a) Note: The latest date a cat is adopted
   (7) cat_availability: {Yes, No}
      (a) Note: Whether the cat is open for adoption
2. Shelters
   (1) shelter_id: unsigned integer
   (2) shelter_name: string
   (3) shelter_location: string
   (4) shelter_email: string
   (5) shelter_phone: 10-digit integer
   (6) shelter_population: integer
   (7) shelter_capacity: integer
      (a) Note: A shelter’s capacity must be greater or equal to the shelter’s population
   (8) shelter_rating: float{1..5}

3. Users:
    (1) user_id: unsigned integer
    (2) user_first_name: string
    (3) user_last_name: string
    (4) user_address: string
    (5) user_email: string
    (6) user_phone: 10-digit integer
4. Owners (subclass of Users)
    (1) user_id: unsigned integer
    (2) owner_cat_count: integer
    (3) owner_rating: float{1..5}
        (a) Note: NULL if not yet rated
    (4) owner_open_to_surrender: {“Yes”, “No”}
5. Adoption Application
    (1) adoption_application_id: unsigned integer
    (2) adoption_application_date: mm/dd/yy hh/mm/ss
    (3) adoption_application_status: {“Submitted”, “Pending”, “Rejected”, “Accepted”}
    (4) adoption_approval_date: mm/dd/yy
        (a) Note: NULL if application status is not “Accepted”
    (5) adoption_date: mm/dd/yy
        (a) Note: cannot be earlier than approval date and NULL if application status is not “Accepted”
6. Surrender Application
    (1) surrender_application_id: unsigned integer
    (2) surrender_reason: {“Don’t want”, “Cannot take care properly”, “Cat is too aggressive”, “Not a good fit”}
    (3) surrender_application_date: mm/dd/yy hh/mm/ss
    (4) surrender_application_status: {“Submitted”, “Pending”, “Rejected”, “Accepted”}
    (5) surrender_approval_date: mm/dd/yy
        (a) Note: NULL if application status is not “Accepted”
    (6) surrender_date: mm/dd/yy
        (a) Note: cannot be earlier than approval date and NULL if application status is not “Accepted”

ii. Verbs
1. House
2. Own
3. Adopt
4. Apply

5. Rate
6. Surrender
7. Connect
8. Create
9. Review

c. Narratives
    i. Cats
        1. Each cat can belong to a shelter or an owner only. A cat cannot belong to neither a shelter nor an owner, however.
        2. Each cat can be applied for by zero or more users
        3. Each cat can have one or more adoption history
        4. A cat can only be available for adoption if it is currently in a shelter or if its owner is open to surrender
        5. A cat can only be surrendered by at most one owner only
    ii. Shelters
        1. Each shelter can house zero or more cats
        2. A shelter can review zero or more applications sent to them
        3. Each shelter must have a rating, calculated by the average of all ratings, including adoption and surrender
        4. Each shelter can receive zero or more applications for adoption or surrender
    iii. Users
        1. A user can apply to adopt one cat at a time from an owner or a shelter, but a user can have multiple applications for different cats
        2. A user can rate zero or more shelters or owners, with the most recent rating taking precedent
        3. A user becomes an owner once they have at least one cat
        4. A user can connect with one or more owners who are open to surrender at least a cat
        5. A user cannot apply for adoption to themself
    iv. Owners
        1. An owner is a subclass of a user
        2. An owner can review zero or more adoption applications sent to them
        3. An owner can create zero or more adoption and/or surrender applications (only shelters accept surrendered cats)
        4. An owner must have at least one cat. An owner reverts to being a user when they surrender all their cats. An owner can surrender zero or more cats
        5. Each owner can own one or more cats not owned by a shelter

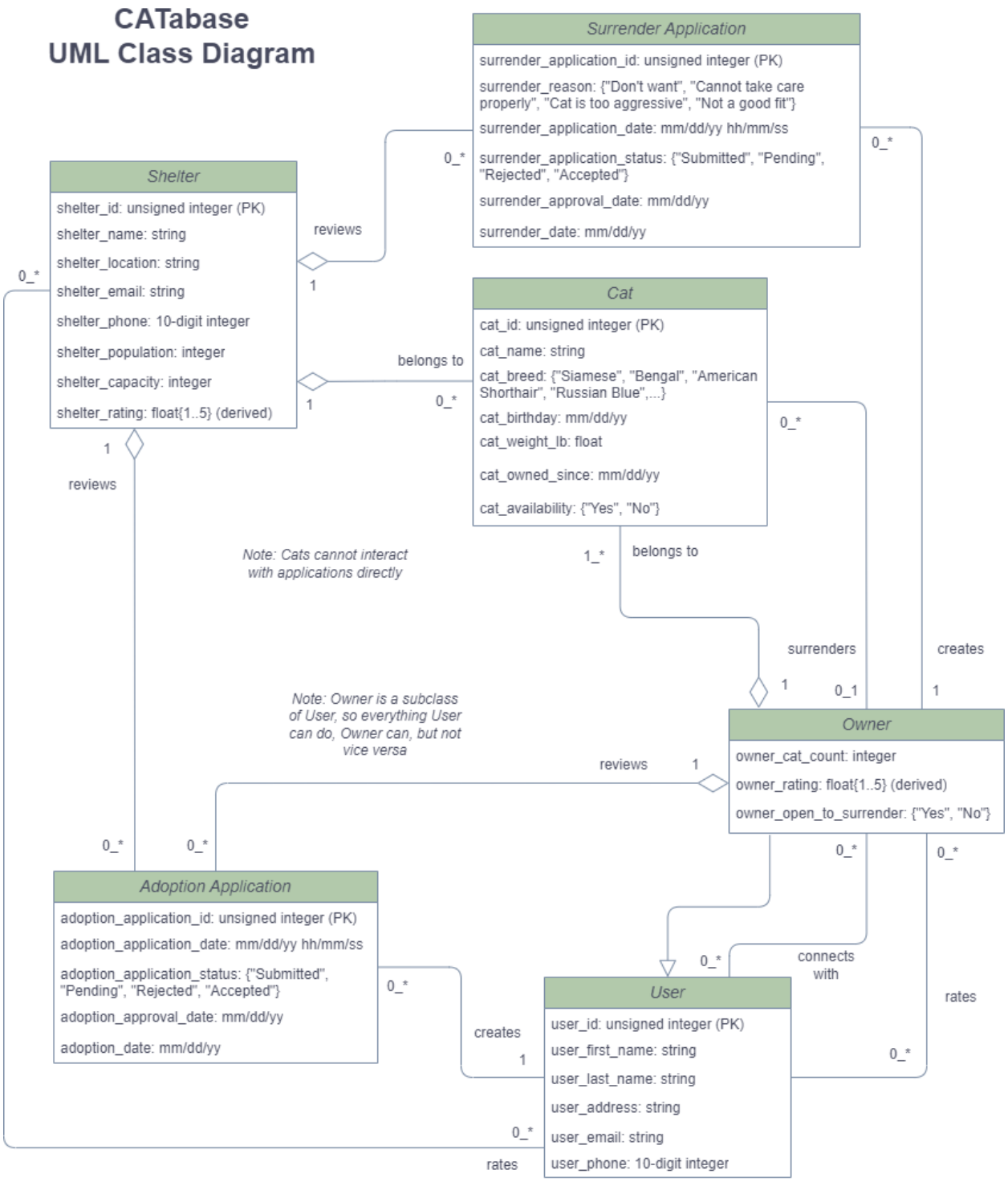
6. Each owner must have a rating
7. An owner open to surrender a cat can connect with one or more users who want to adopt
8. An owner can rate zero or more shelters or other owners, with the most recent rating taking precedent

v. Adoption Application
    1. An adoption application can be submitted to and reviewed by one owner or shelter

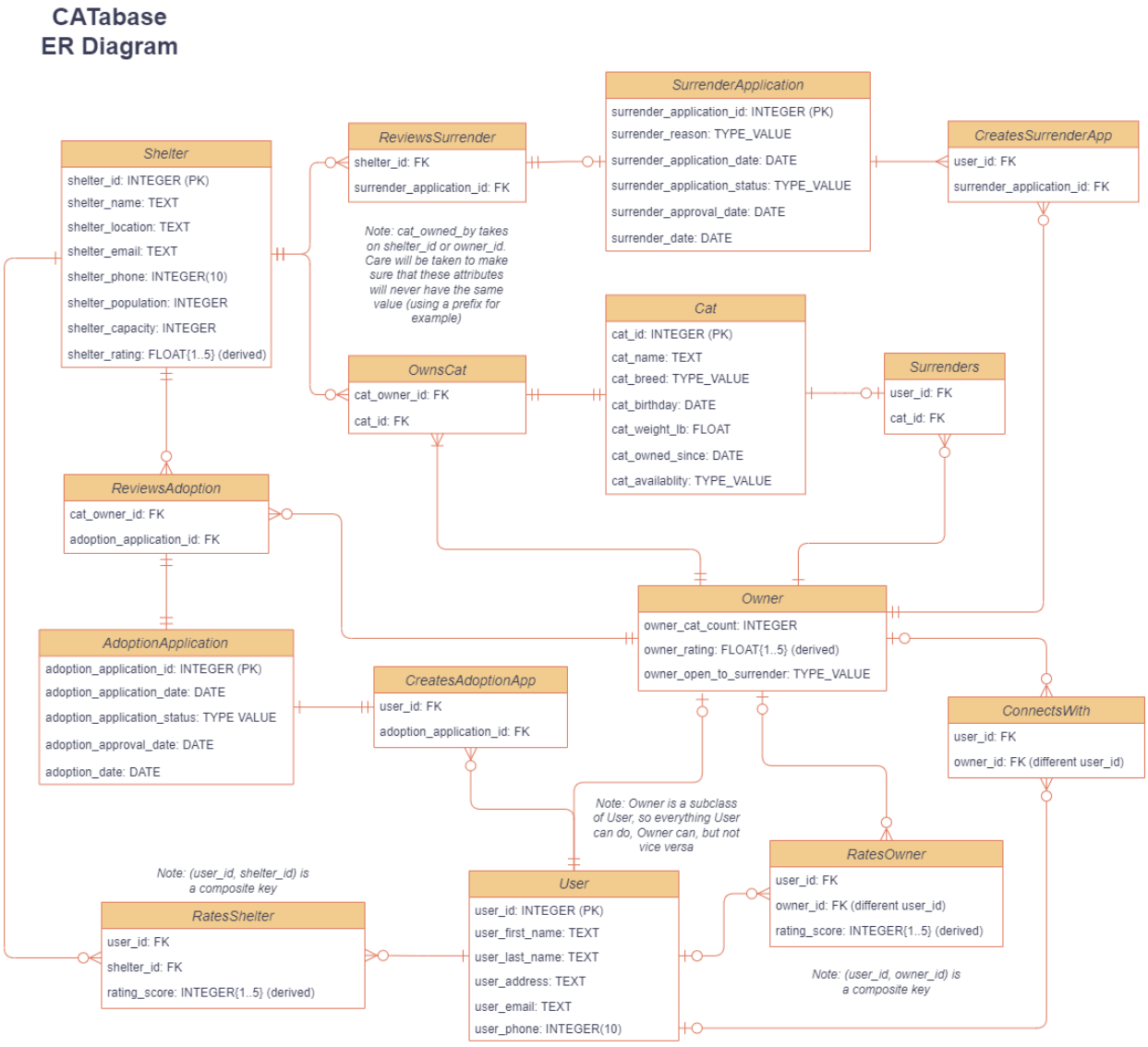
vi. Surrender Application
    1. A surrender application can be submitted to and reviewed by one owner or shelter

d. Challenges
    i. Should a cat have multiple owners?
    ii. Should a general user (non-owner) have a rating? How should it be handled if an owner reverts back to being a user?
    iii. Should adoption history be a separate table?

II. Conceptual Model (UML)



III. Entity-Relationship Model (ERD)



IV. Relational Schema in BCNF

a. Schema with Functional Dependencies

i. Cat(cat_id, cat_name, cat_breed, cat_birthday, cat_weight_lb, cat_owned_since, cat_availability)
   1. cat_id → {cat_name, cat_breed, cat_birthday, cat_weight_lb, cat_owned_since, cat_availability}
   2. Candidate key: cat_id

ii. Shelter(shelter_id, shelter_name, shelter_location, shelter_email, shelter_phone, shelter_population, shelter_capacity, shelter_rating)
   1. shelter_id → {shelter_name, shelter_location, shelter_email, shelter_phone, shelter_population, shelter_capacity, shelter_rating}

iii. User(user_id, user_first_name, user_last_name, user_address, user_email, user_phone)
   1. user_id → {user_first_name, user_last_name, user_address, user_email, user_phone}

iv. Owner(user_id, owner_cat_count, owner_rating, owner_open_to_surrender)
   1. user_id → {{User}, owner_cat_count, owner_rating, owner_open_to_surrender}

v. AdoptionApplication(adoption_application_id, adoption_application_date, adoption_application_status, adoption_approval_date, adoption_date)
   1. adoption_application_id → adoption_application_date, adoption_application_status, adoption_approval_date, adoption_date

vi. SurrenderApplication(surrender_application_id, surrender_reason, surrender_application_date, surrender_application_status, surrender_approval_date, surrender_date)
   1. surrender_application_id → surrender_reason, surrender_application_date, surrender_application_status, surrender_approval_date, surrender_date

vii. OwnsCat(*cat_owner_id, cat_id*).
   1. *Note: cat_owner_id takes on the values of user_id (owner) or shelter_id. Care will be taken to ensure these values are unique.*
   2. Candidate key: {*cat_owner_id, cat_id*}

viii. ConnectsWith(*user_id, owner_id)*
   1. Candidate key: {*user_id, owner_id*}

ix. RatesOwner(*user_id, owner_id,* rating_score)
   1. {*user_id, owner_id*} → rating score
   2. *Note:* each attribute of the composite key are themselves keys in different tables

x. RatesShelter(*user_id, shelter_id,* rating_score)
    1. {*user_id, shelter_id*} → rating score
    2. *Note:* each attribute of the composite key are themselves keys in different tables

xi. Surrenders(*user_id, cat_id)*
    1. Candidate key: {*user_id, cat_id*}

xii. CreatesSurrenderApp(*user_id, surrender_application_id*)
    1. Candidate key: {*user_id, surrender_application_id*}

xiii. ReviewsSurrender(*shelter_id, surrender_application_id*)
    1. Candidate key: {*shelter_id, surrender_application_id*}

xiv. CreatesAdoptionApp(*user_id, adoption_application_id*)
    1. Candidate key: {*user_id, adoption_application_id*}

xv. ReviewsAdoption(*cat_owner_id, adoption_application_id*)
    1. Candidate key: {*cat_owner_id, adoption_application_id*}

b. BCNF Analysis

i. Schema is in 1NF:
    1. No multi-valued attributes

ii. Schema is in 2NF:
    1. Schema is in 1NF
    2. No partial functional dependencies; in all tables, all proper subsets of the candidate key give non-key attributes

iii. Schema is in 3NF:
    1. Schema is in 2NF
    2. No transitive dependency: no tables exhibit any transitive dependency

iv. Schema is in BCNF
    1. Schema is in 3NF
    2. For every dependency A → B in the schema, A is either a candidate key or a super key. Therefore, the Relational Schema is in BCNF