**Part I PROPOSAL**

1. **What problems are you solving?**
   1. Address problems that are difficult to quickly track and update status and location when animals enter or leave the shelter.
   2. Failure to effectively track and manage animal health conditions and medical records will address regular medical care and difficulties in responding to urgent medical situations.
2. **How will your features solve them?**
   1. Tracking Animal Status and Location:
      * Providing a way to easily update the relevant tables when an animal enters or leaves the shelter to promptly track and update the status and location of animals.
   2. Managing Animal Health Status and Medical Records:
      * Effectively tracking and managing the health status and medical records of animals to prepare for regular medical care and emergencies.
3. **What are your tables, how are they composed, and why is that justified by your solutions?**
   1. Tracking Animal Status and Location:
      * Table Name: Animal
      * Columns: AnimalID (Primary key), Status (Animal status), Location (Animal location), EntryDate (Date of entry), ExitDate (Date of exit), etc.
      * This table efficiently records basic information about the animal's status and location, allowing easy updates when an animal enters or leaves the shelter.
   2. Managing Animal Health Status and Medical Records:
      * Table Name: MedicalRecords
      * Columns: RecordID (Primary key), AnimalID (Animal identifier/foreign key), HealthStatus (Health status), MedicalHistory (Medical history), etc.
      * This table provides a detailed record of the animal's health status and medical history, facilitating regular health management and preparation for emergency medical situations.
4. **What are your database tools (procedures, functions, etc.), and why are they justified?**
   1. Tracking Animal Status and Location:
      * Stored Procedure for Updating Animal Status
        1. Purpose: To provide a centralized and controlled way of updating animal status and location.
        2. Justification: This stored procedure ensures data consistency by handling the update process in a secure and controlled manner, preventing unauthorized changes.
      * Trigger for Logging Changes
        1. Purpose: To log any changes made to the `Animal` table.
        2. Justification: This trigger helps in maintaining an audit trail by recording modifications, providing accountability and facilitating error tracking.
   2. Managing Animal Health Status and Medical Records:
      * Stored Function for Retrieving Medical History:
        1. Purpose: To retrieve the medical history of a specific animal.
        2. Justification: This stored function enhances usability by encapsulating the logic for retrieving medical history, making it easier to integrate into queries and reports.
      * View for Aggregated Health Statistics:
        1. Purpose: To provide an aggregated view of health statistics, such as the number of animals with a specific health status.
        2. Justification: This view simplifies reporting and analysis, offering a consolidated perspective on health-related data without the need for complex queries.

Table: animal

1. AnimalID (Primary Key, INT)
2. Species (VARCHAR)
3. Name (VARCHAR)
4. Age (INT)
5. Status (VARCHAR)
6. Location (VARCHAR)
7. EntryDate (DATETIME)
8. ExitDate (DATETIME)

Constraints:AnimalID is the primary key to uniquely identify each animal.

Table: medicalRecord

1. RecordID (Primary Key, INT)
2. AnimalID (Foreign Key referencing `animal.AnimalID`, INT)
3. HealthStatus (VARCHAR)
4. MedicalHistory (TEXT)

Constraints:

RecordID is the primary key for uniquely identifying each medical record.

AnimalID is a foreign key linking each medical record to a specific animal in the animal table.