

Digital Zoo Project - Week 2

Abdellah Sabhi
Jalal Mammadov
Hazal



Project Overview

- Main Task:

In Week 2, we focused on enhancing functionality and improving the user experience of our zoo management system.

- Week 2 Objective:

Set up the **backend** using Django & Django REST Framework (DRF).

Develop API endpoints for **animals, species, habitats, feeding schedules, and tickets**.

Implement **user authentication** (registration & login).

Connect the **frontend (React.js)** with the **backend**.

Features:

User Interaction:

- Dropdown menus for species and habitats now pull data dynamically from the database.
- Forms prevent incorrect data entry.

Animal & Habitat Management:

- UI improvements ensure clear visibility of animals and their linked habitats.
- Users can now correctly assign species and habitats to animals.

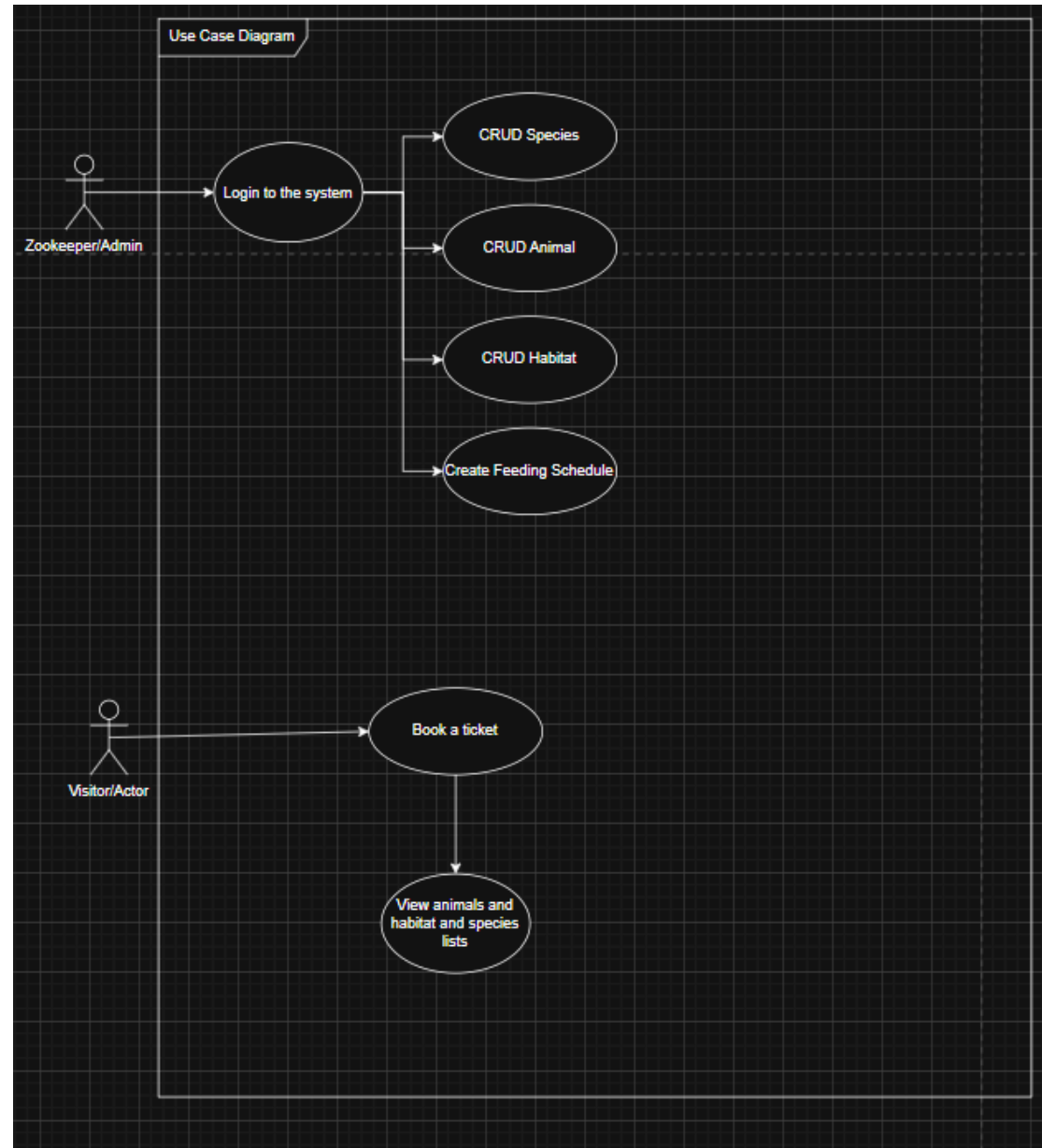
Feeding Schedule Management:

- Feeding schedules now fetch and display correctly.
- Improved layout for better readability.

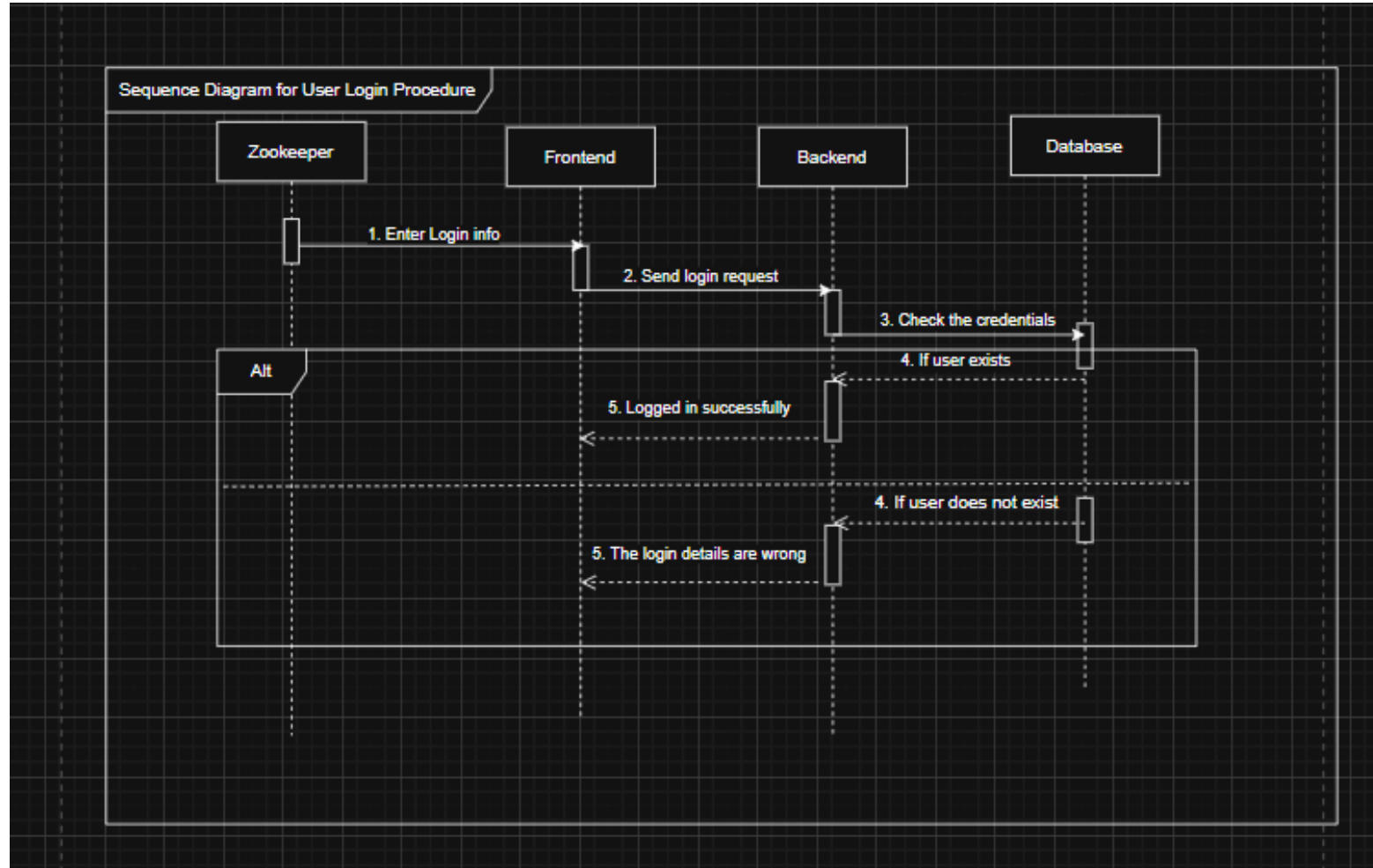
Ticketing System:

- Tickets now store correct visitor names and expiry dates.
- Improved validation to prevent invalid ticket creation.

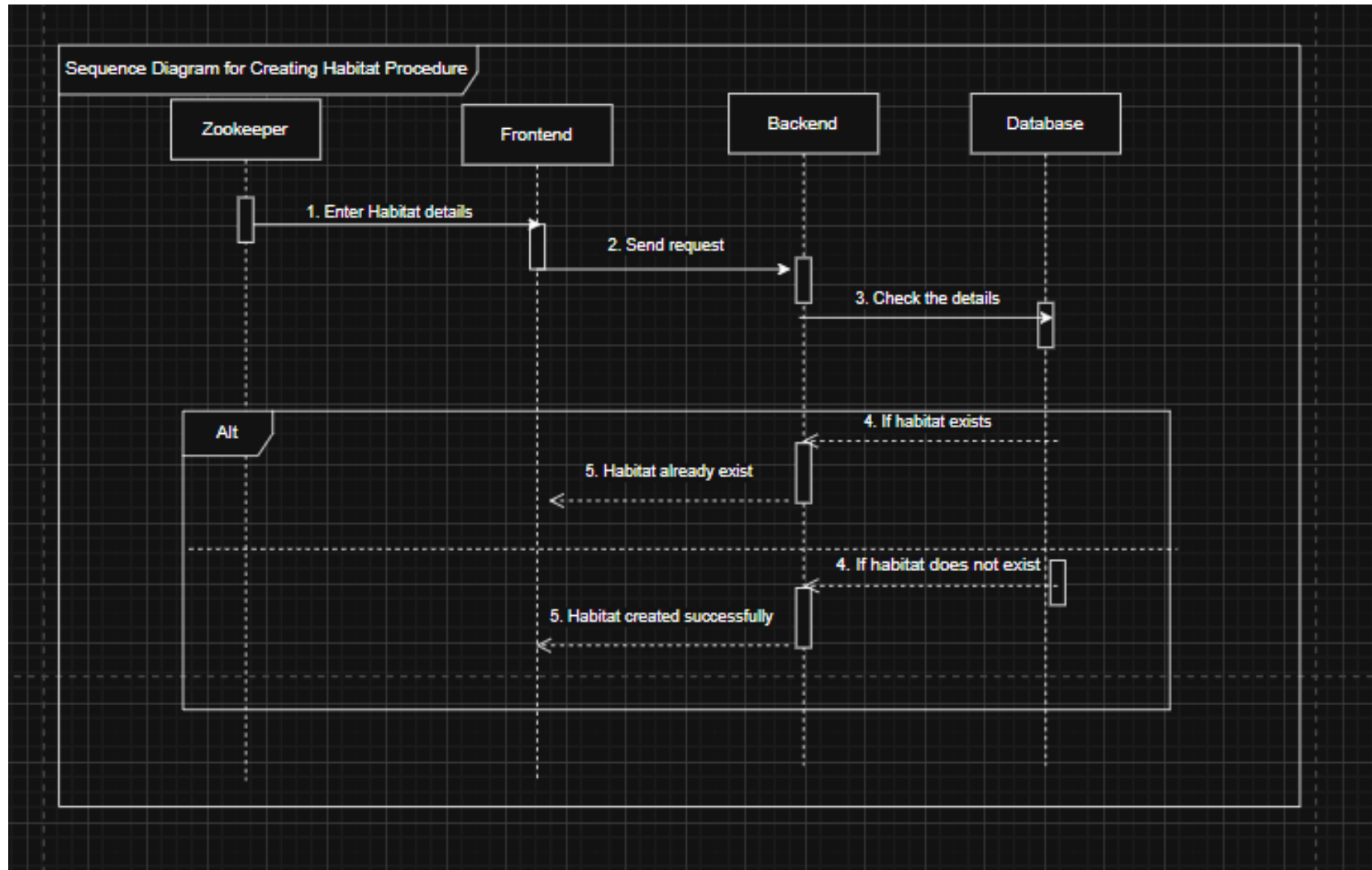
Use case Diagram:



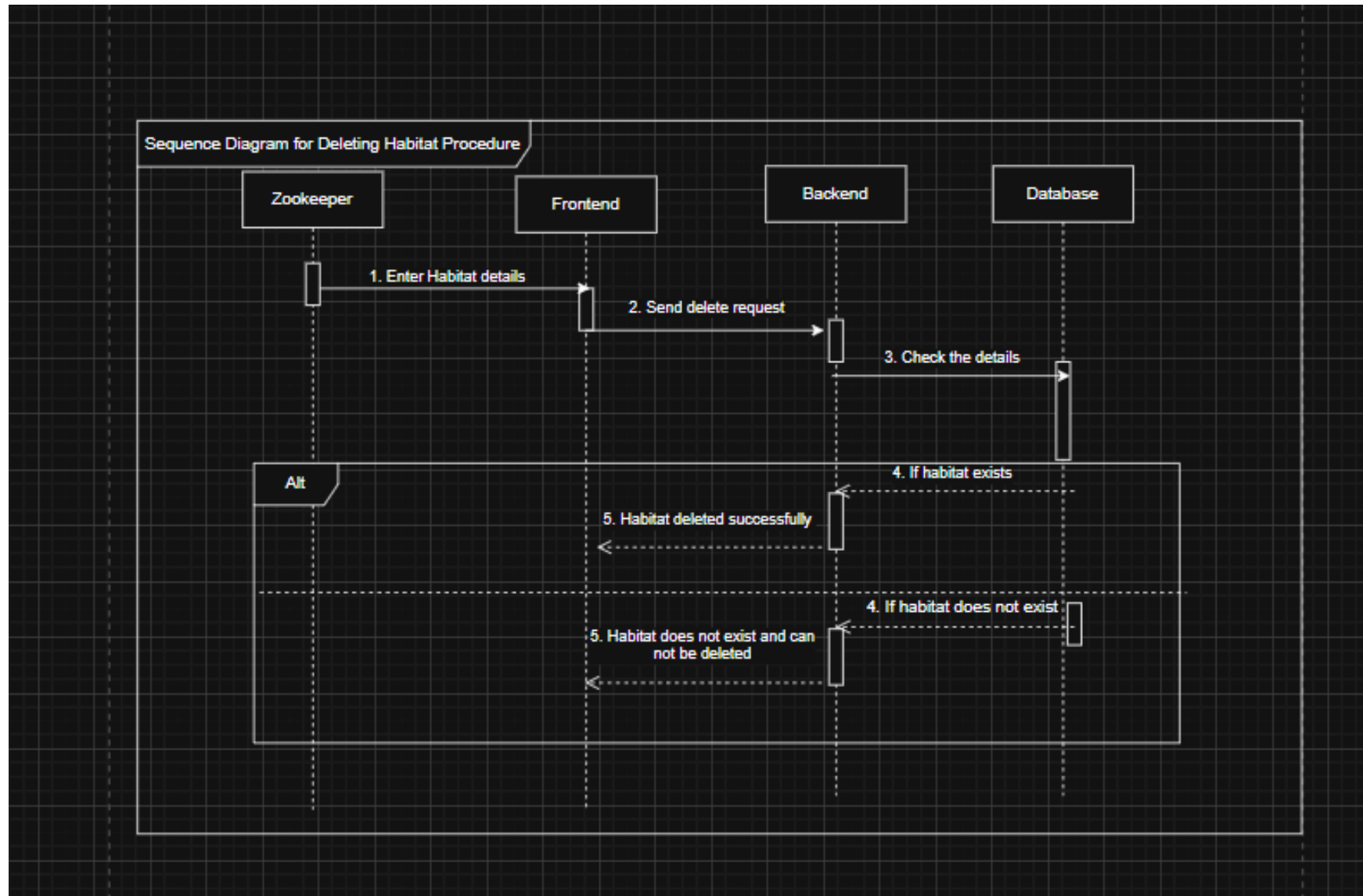
Sequence diagram



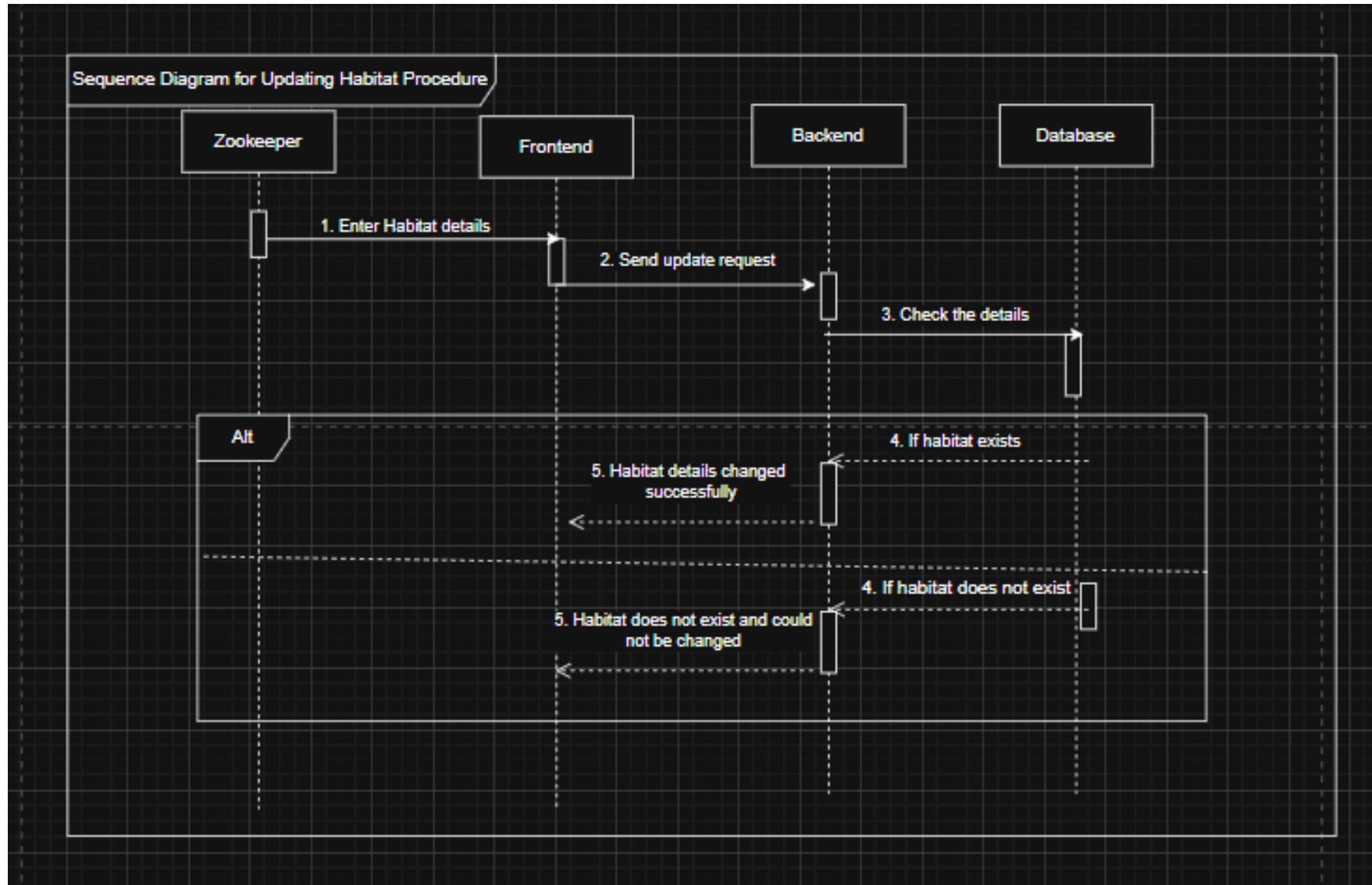
Sequence diagram



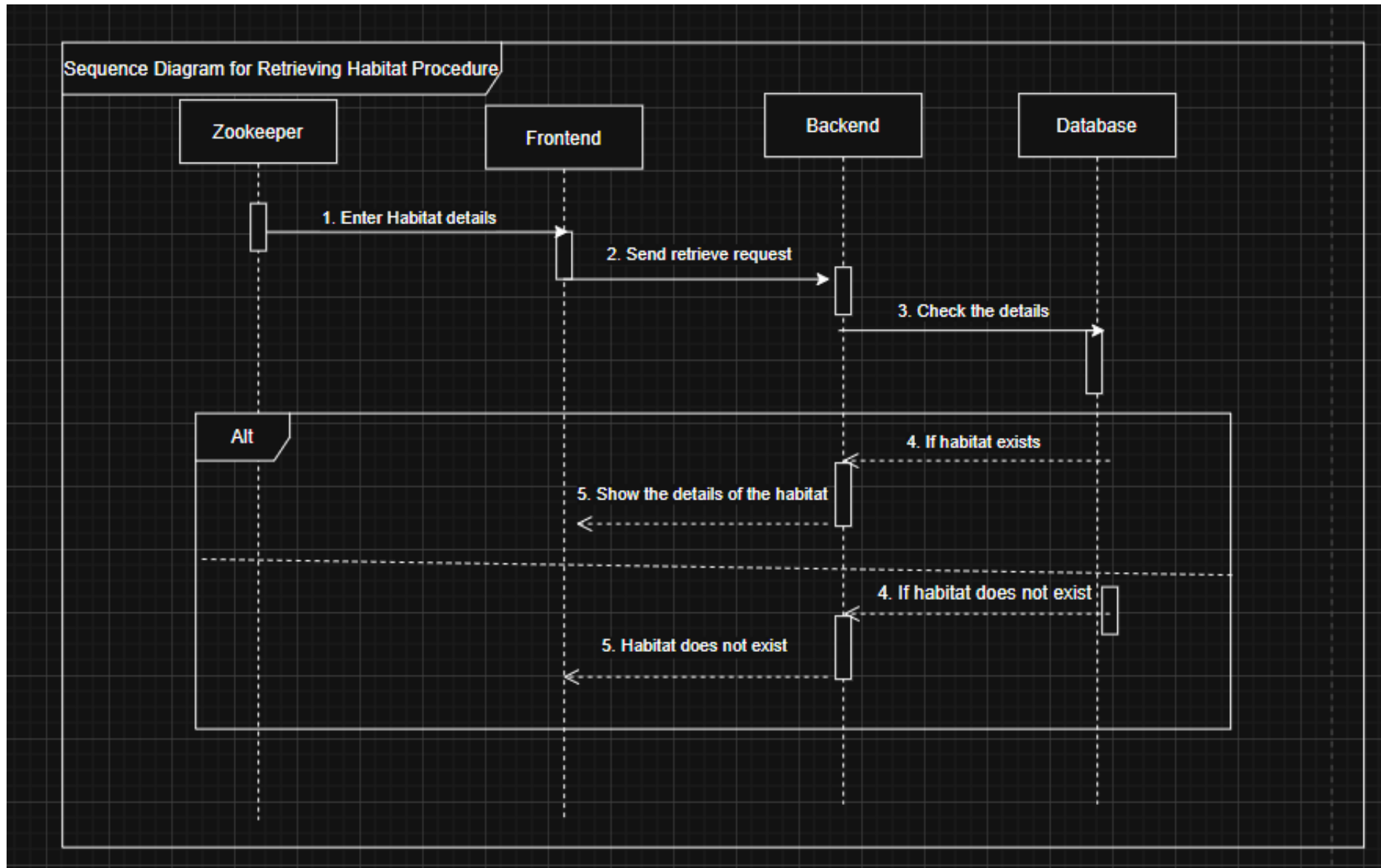
Sequence diagram



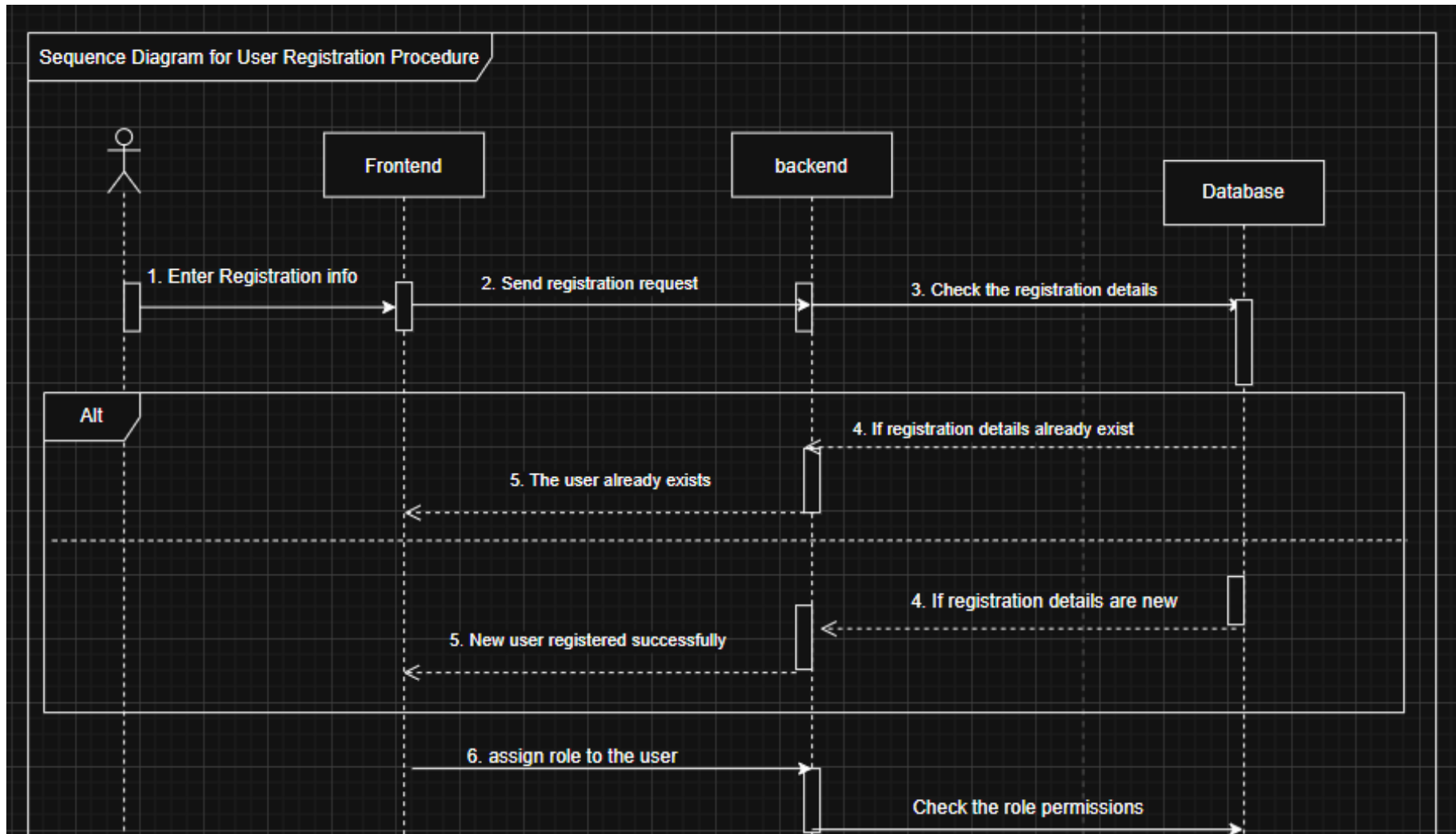
Sequence diagram



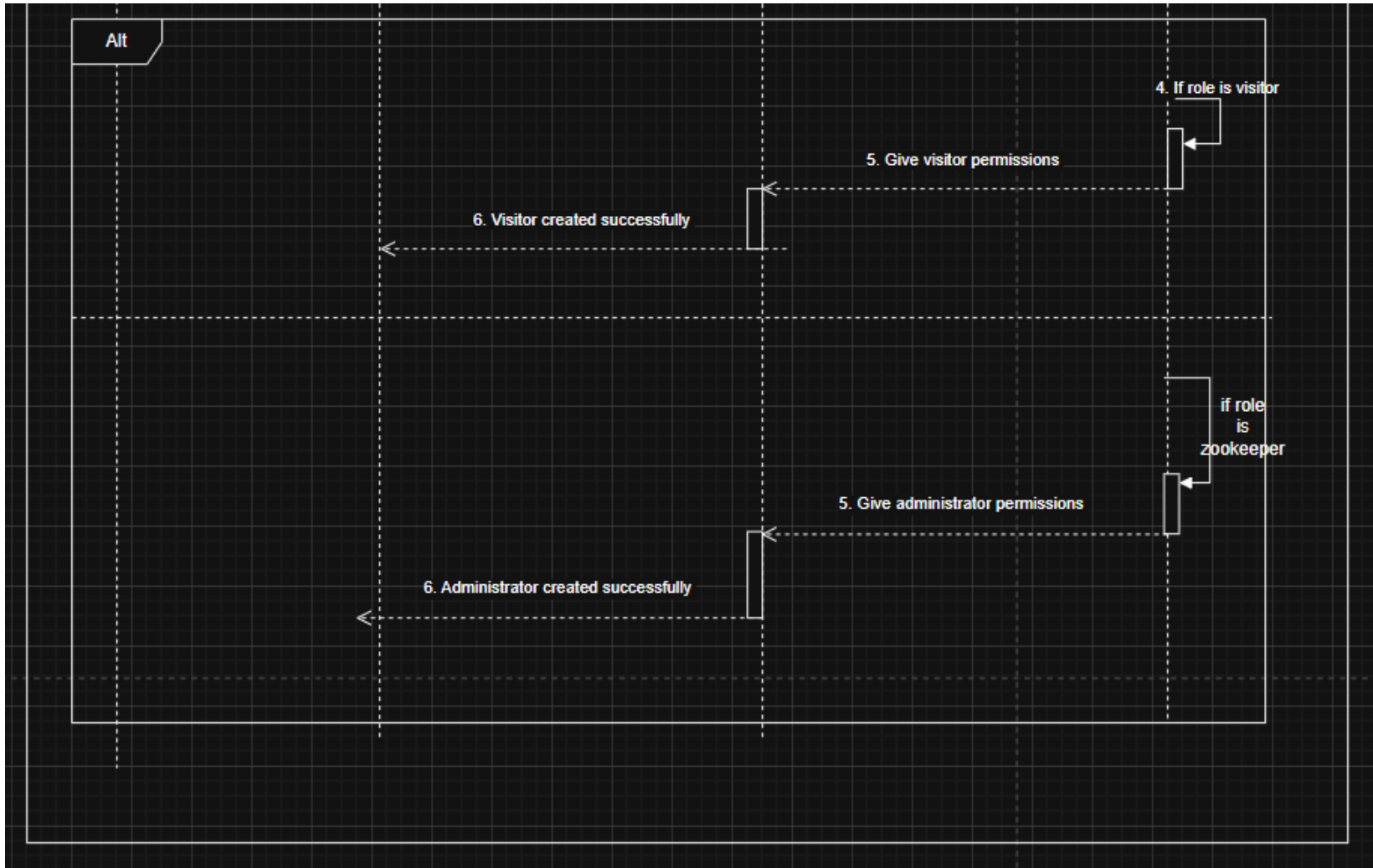
Sequence diagram



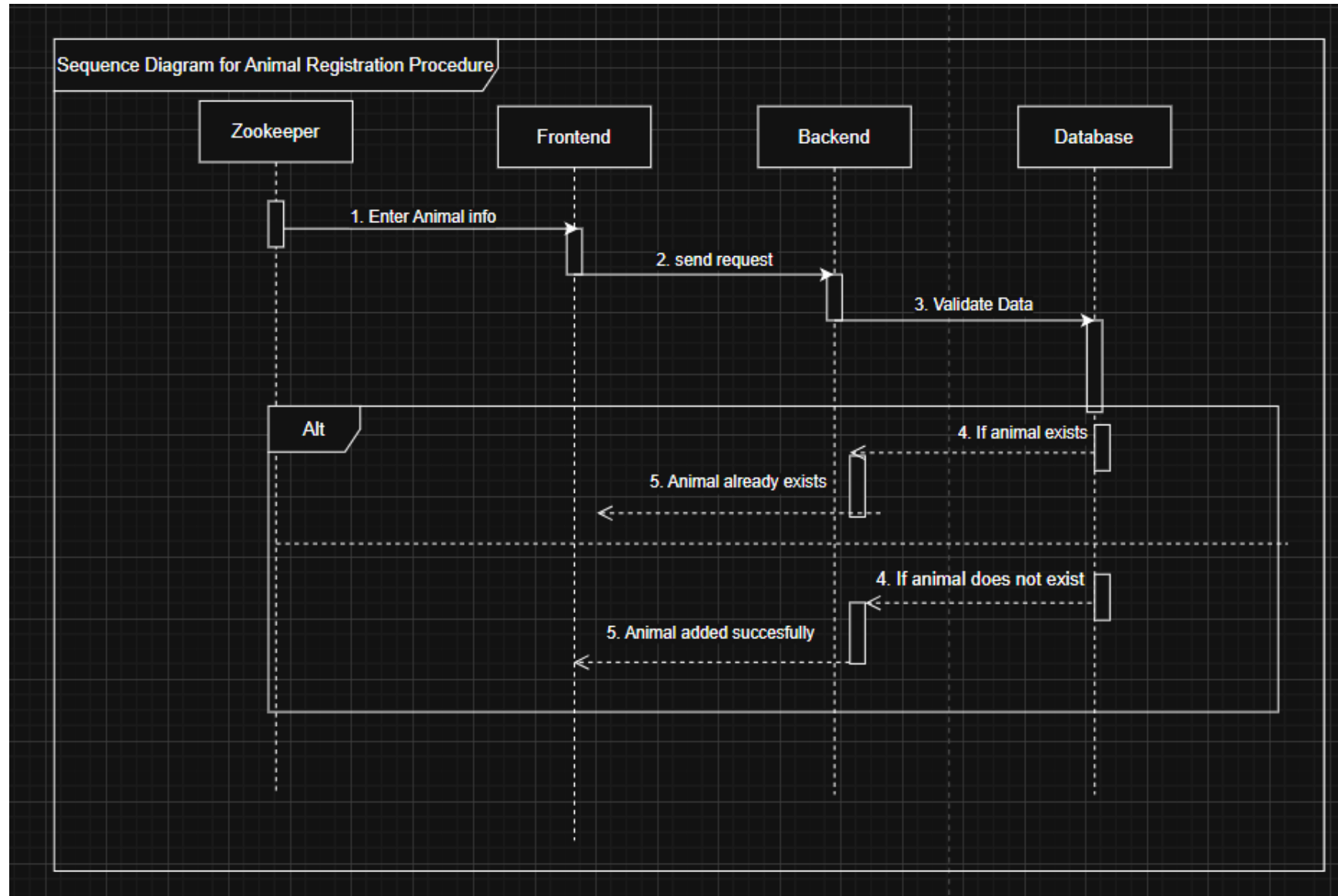
Sequence diagram



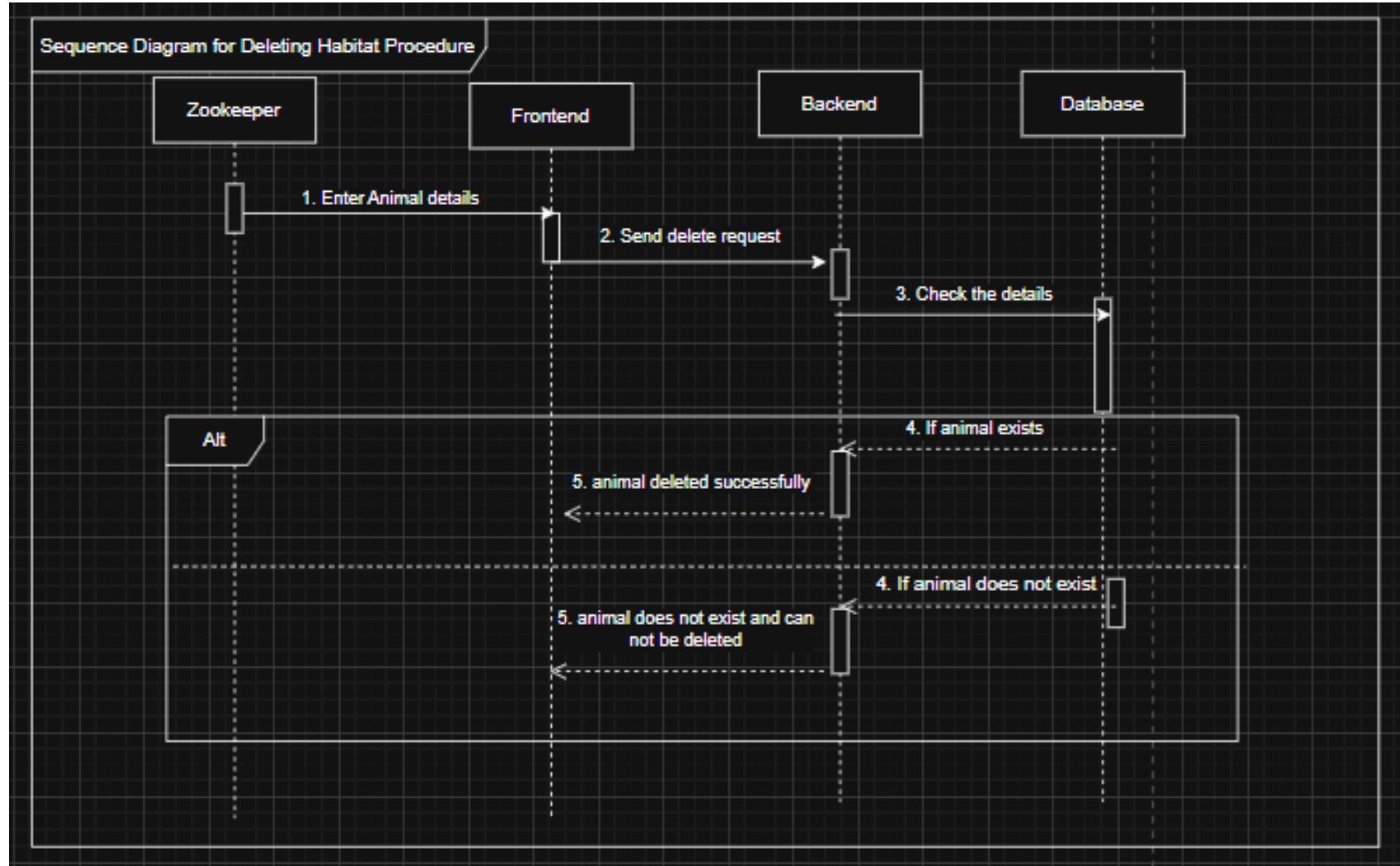
Sequence diagram



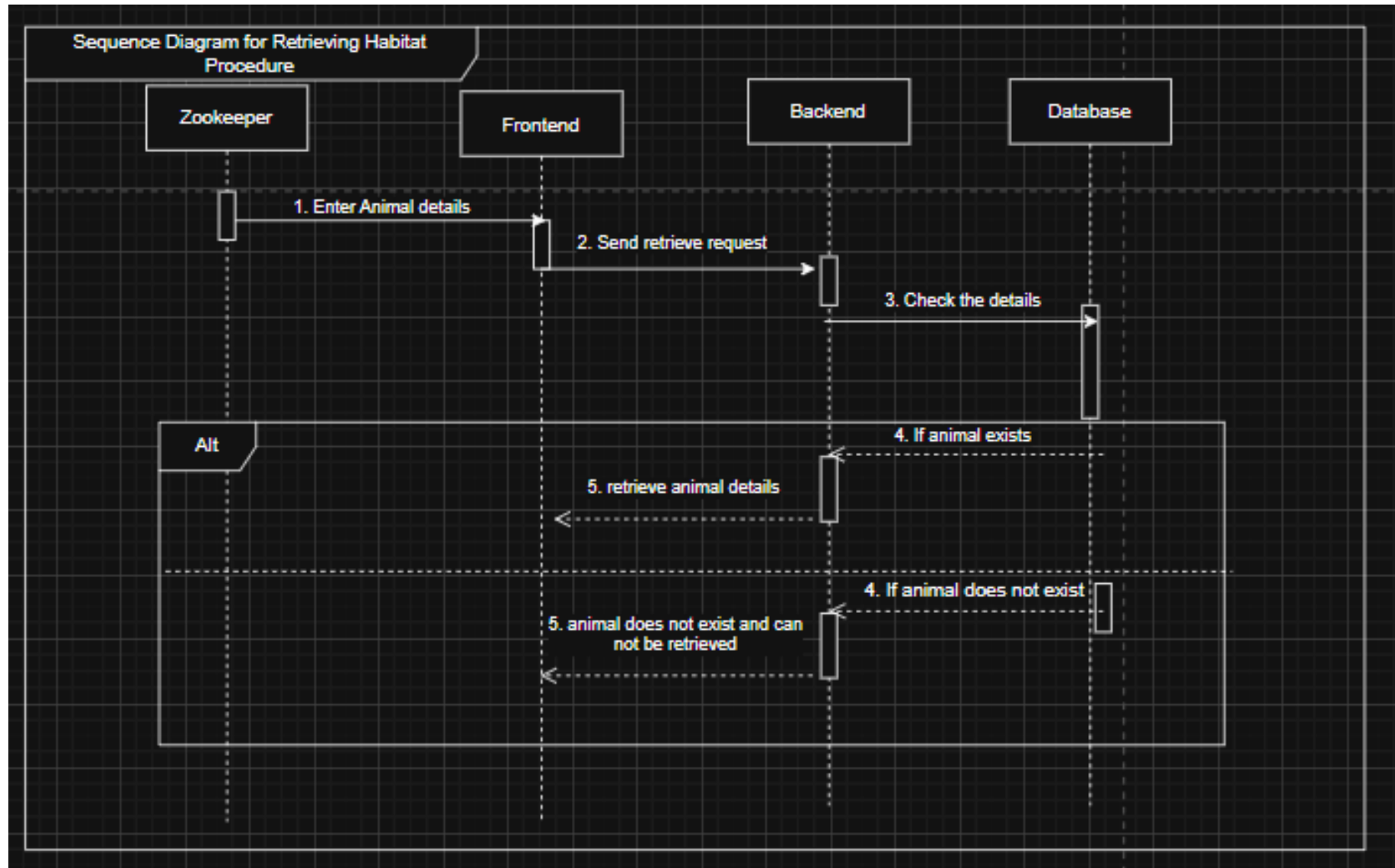
Sequence diagram



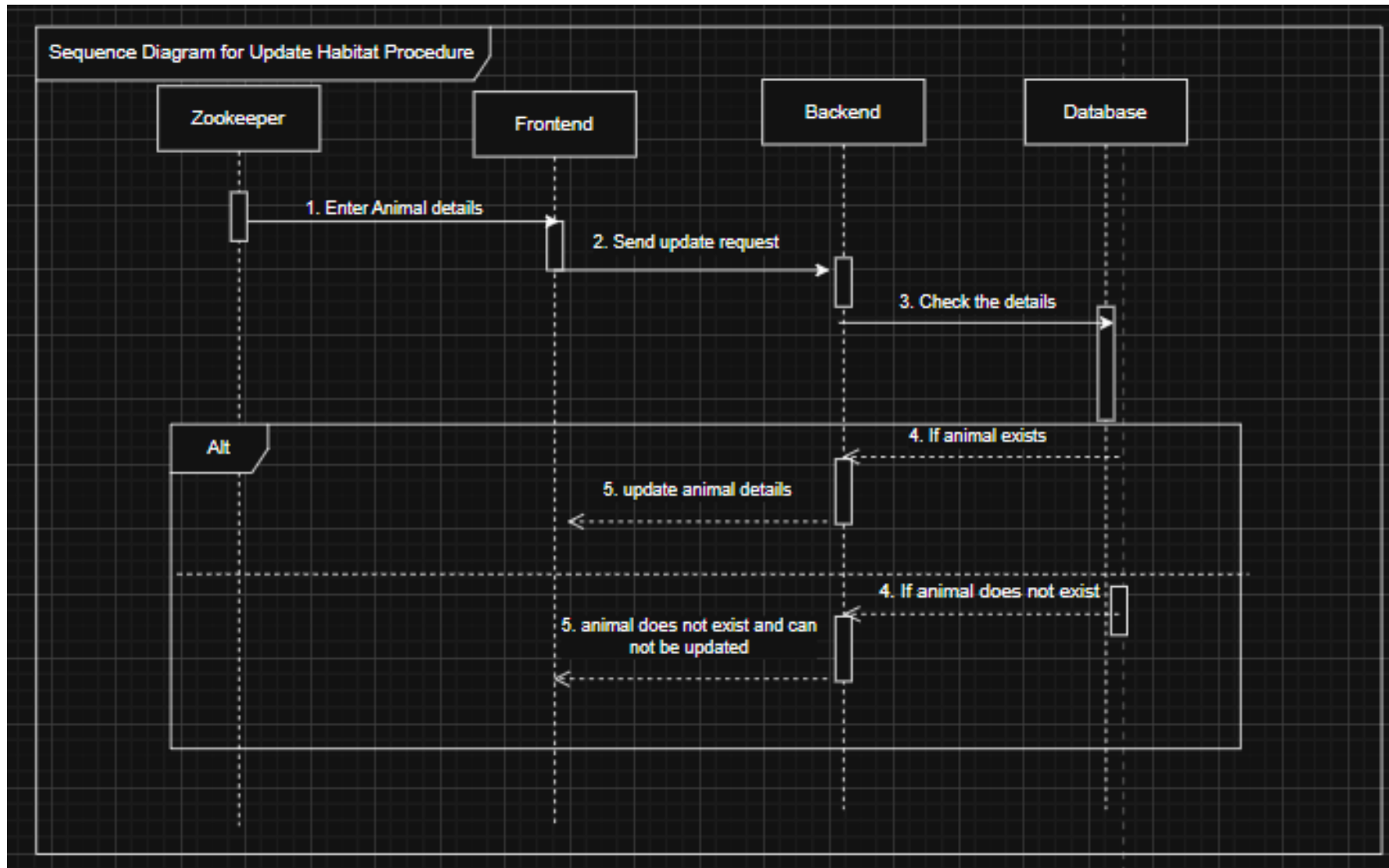
Sequence diagram



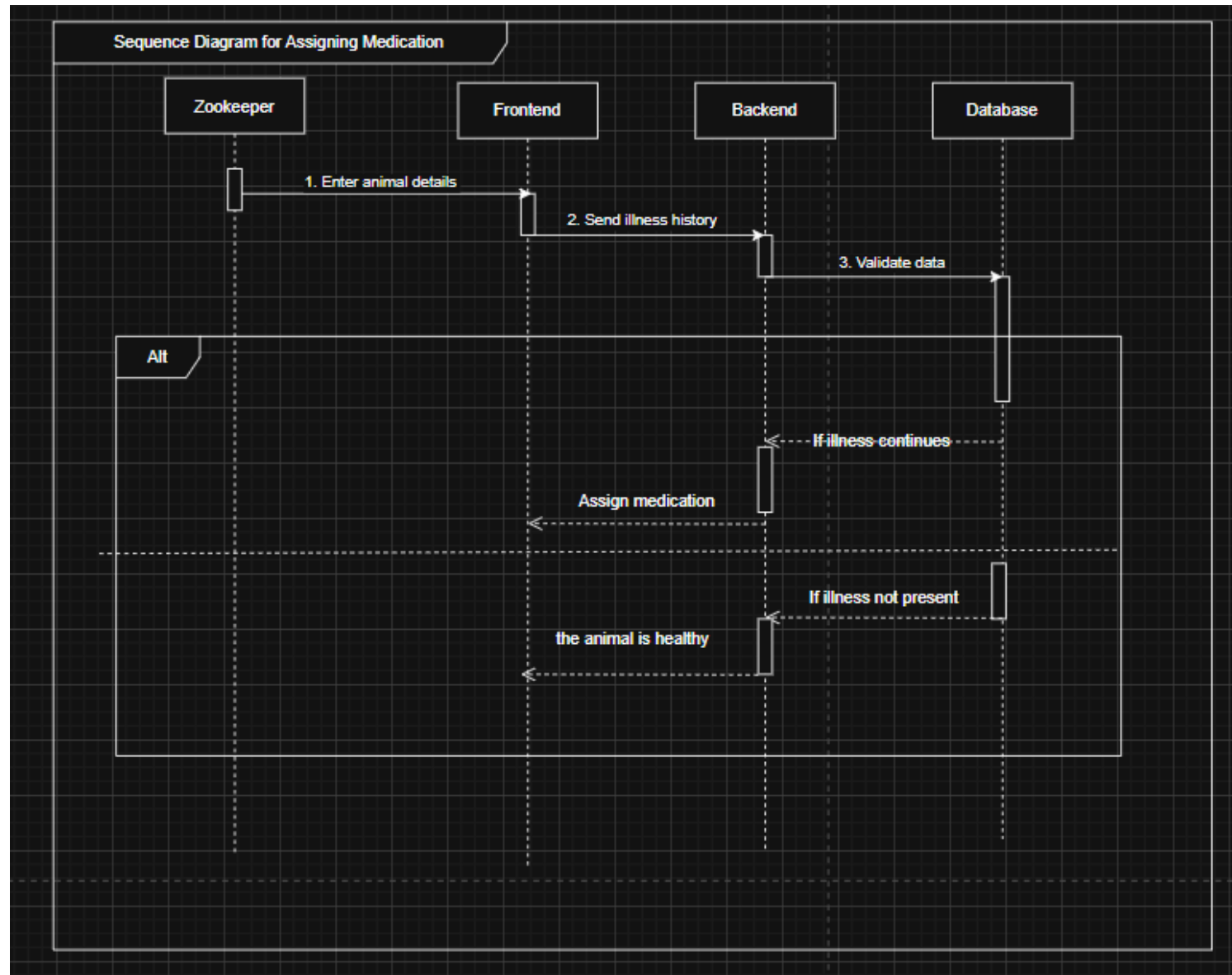
Sequence diagram



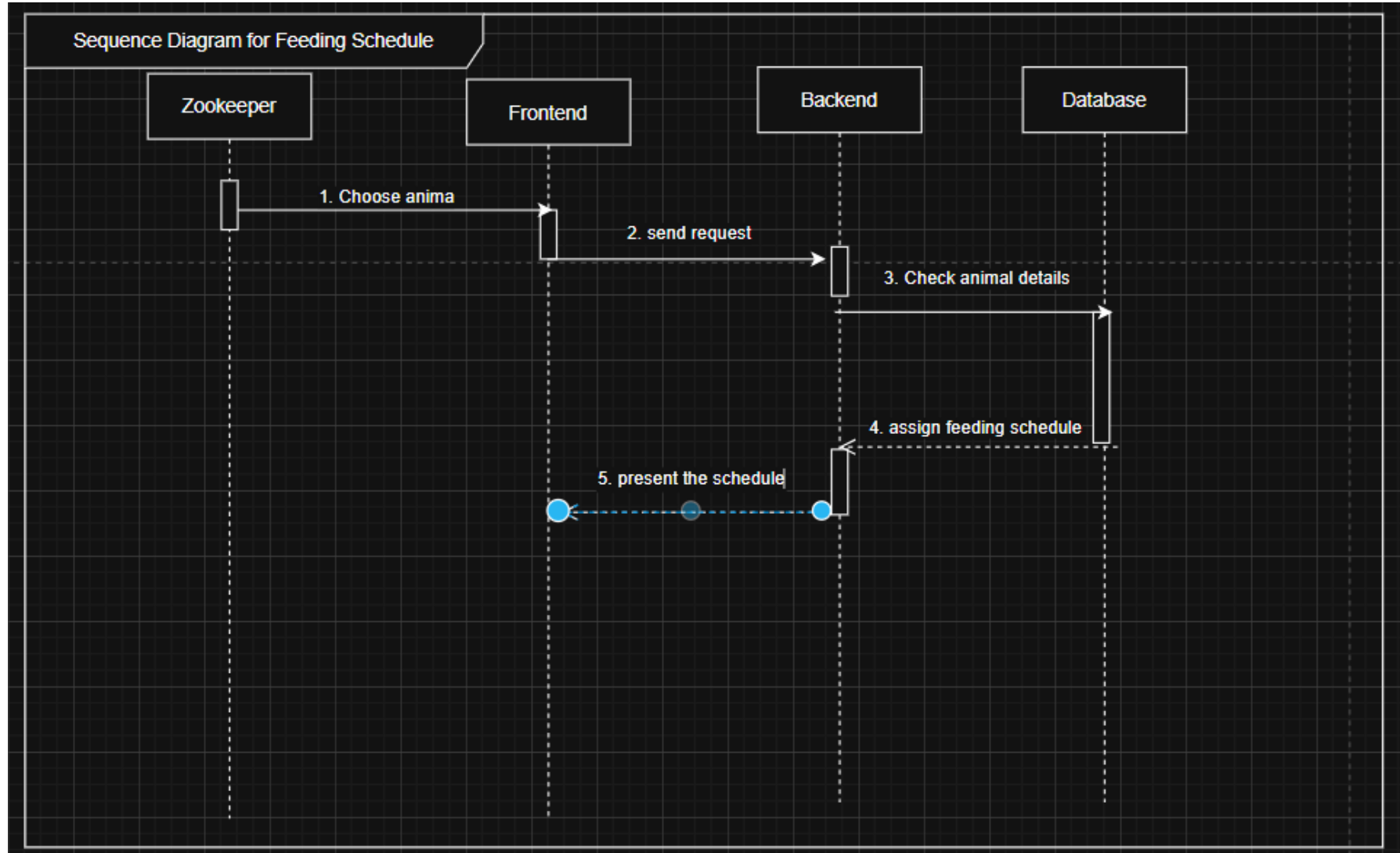
Sequence diagram



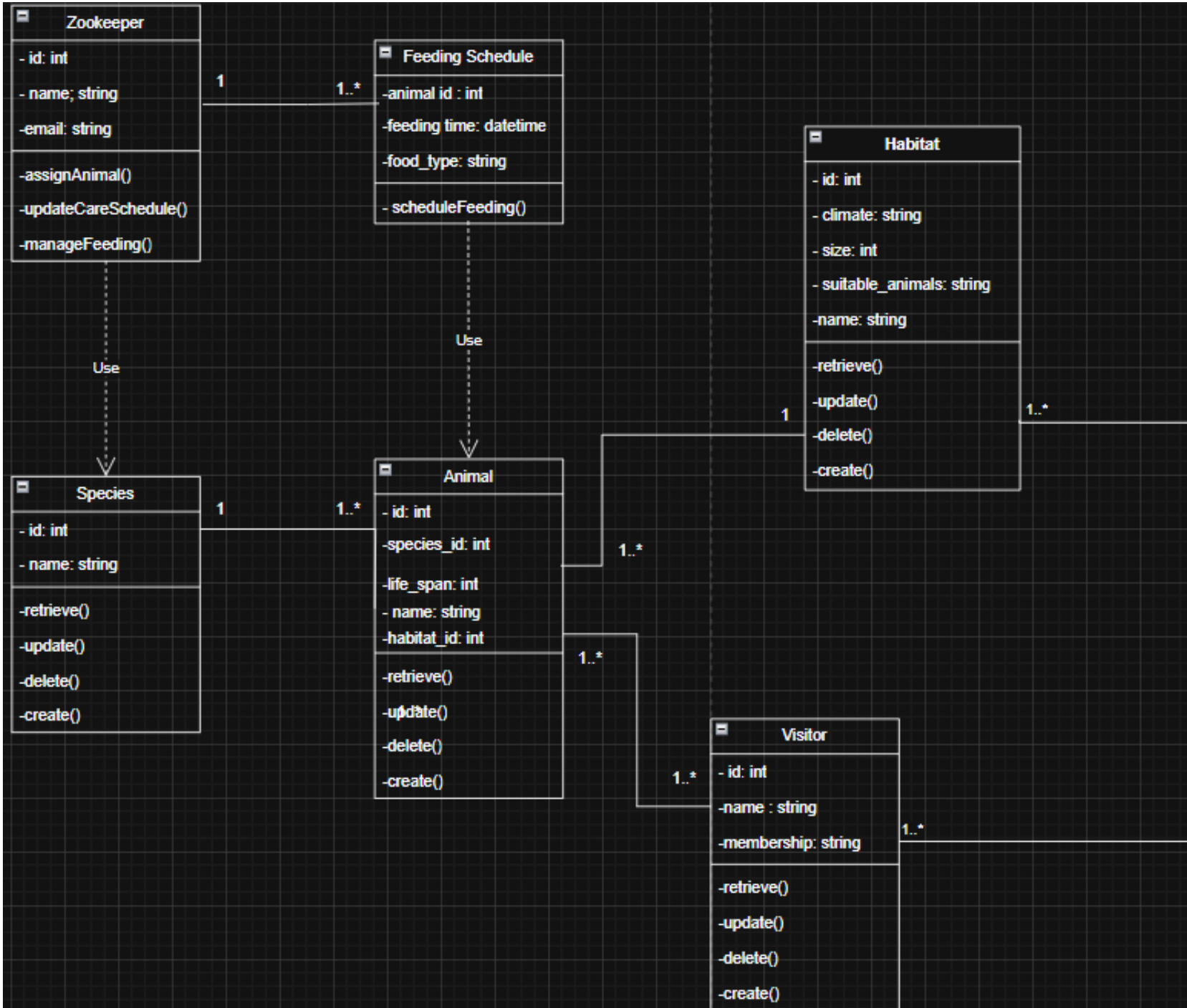
Sequence diagram



Sequence diagram




Class diagram



Challenges

Dropdowns for species and habitats were initially empty due to incorrect API calls.

White background made forms and buttons difficult to see.



Next Steps

- Fix authentication and authorization issues.
 - Add error handling for missing or undefined states.
 - Implement data validation & user role permissions.
- 