System Requirements Specification Index

For

Dog Daycare Management System

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

1 PROJECT ABSTRACT

Paws & Play Dog Daycare requires a management system to digitize their operations. The system will track dogs, owners, and daily activities at the daycare. It will enable staff to efficiently manage dog registrations, check-ins/check-outs, activity scheduling, and maintain records of each dog's behavior and preferences. This system provides an organized way for the daycare to manage their canine clients and provide personalized care.

2 BUSINESS REQUIREMENTS:

Screen Name	Console input screen
Problem Statement	1. System needs to store and manage different types of data (dogs, owners, activities) 2. System must support operations such as dog registration, check-in/check-out, and activity assignment 3. Console should implement object-oriented concepts like inheritance and method overriding to achieve desired outcome

3 CONSTRAINTS

3.1 CLASS REQUIREMENTS

1. `Dog` Class:
   - Attributes: dog_id, name, breed, age, weight, is_checked_in
   - Methods: display_info(), check_in(), check_out()

- Example: `Dog("D001", "Buddy", "Golden Retriever", 3, 65.5, False)`

2. `Owner` Class:
   - Attributes: owner_id, name, email, phone, dogs_registered
   - Methods: display_info(), register_dog(), pickup_dog()
   - Example: `Owner("O001", "John Smith", "john@example.com", "555-1234", [])`
3. `SmallDog` Class (inherits from `Dog`):
   - Additional attributes: toy_preference
   - Override methods: display_info()
   - Example: `SmallDog("D002", "Daisy", "Yorkshire Terrier", 5, 7.5, False, "Plush toys")`
4. ‘LargeDog` Class (inherits from `Dog`):
   - Additional attributes: exercise_needs
   - Override methods: display_info()
   - Example: `LargeDog("D003", "Max", "German Shepherd", 2, 75.0, False, "High")`
5. `Daycare` Class:
   - Attributes: name, address, dogs, owners, available_activities
   - Methods: add_dog(), add_owner(), check_in_dog(), check_out_dog(), get_checked_in_dogs(), search_dog_by_name(), search_dog_by_breed()
   - Static methods: get_dog_count(), get_owner_count()
   - Example: `Daycare("Paws & Play", "456 Park Ave, Dogtown")`

3.2 Operation Constraints

1. Dog Check-in:
   - Owner must exist in the system
   - Dog must exist in the system
   - Dog must not already be checked in
   - Must update dog check-in status
2. Dog Check-out:
   - Dog and owner must exist in the system
   - Dog must be currently checked in
   - Must update dog check-in status

3. Owner Registration:
   - Owner ID must be unique
   - Email must be valid format (must contain @ and a domain)
   - Phone must be in valid format (###-###-####)
4. Dog Addition:
   - Dog ID must be unique
   - Age must be positive
   - Weight must be positive
   - Dog must be assigned to the correct subclass (SmallDog/LargeDog)
5. Exception Handling:
   - Must handle DogNotFoundException
   - Must handle OwnerNotFoundException
   - Must handle DogAlreadyCheckedInException
   - Must handle InvalidInputException
   - Must handle DogNotCheckedInException
6. Object-Oriented Requirements:
   - Must use proper encapsulation (private attributes with getters/setters)
   - Must implement inheritance for dog types
   - Must use polymorphism with method overriding
   - Must implement static methods and class variables

3.3 OUTPUT CONSTRAINTS

1. Display Format:
   - Dog information: display ID, name, breed, age, weight, check-in status
   - Owner information: display ID, name, email, phone, number of dogs registered
   - Each item must be displayed on a new line with proper formatting
2. Output Format:
   - Must show in this order:
     - Show "== DOG DAYCARE MANAGEMENT SYSTEM =="
     - Show "Daycare Name: {name}"

- Show "Address: {address}"
- Show "Total Dogs: {count}"
- Show "Total Owners: {count}"
- Show "Current Dogs in Daycare:"
- Show dogs with format: "{id} | {name} ({breed}) | {age} years | {weight} lbs | Status: {status}"
- Show "Search Results:" when displaying search results

4. TEMPLATE CODE STRUCTURE:

1. Dog Classes:
   - \`Dog\` (base class)
   - \`SmallDog\` (derived class)
   - \`LargeDog\` (derived class)

2. Owner Class:
   - \`Owner\`

3. Daycare Class:
   - \`Daycare\`

4. Exception Classes:
   - \`DogNotFoundException\`
   - \`OwnerNotFoundException\`
   - \`DogAlreadyCheckedInException\`
   - \`DogNotCheckedInException\`
   - \`InvalidInputException\`

5. Program Control:
   - \`main()\` - main program function

5. EXECUTION STEPS TO FOLLOW:

1. Run the program
2. View the main menu
3. Select operations:
   - Option 1: Add New Dog

   - Option 2: Add New Owner
   - Option 3: Check-in Dog
   - Option 4: Check-out Dog
   - Option 5: Display All Dogs
   - Option 6: Display All Owners
   - Option 7: Search for Dogs
   - Option 0: Exit
4. Perform operations on the daycare system
5. View results after each operation
6. Exit program when finished