

# **Project Name**

**Virtual Pet Care System with Daily Reminders**

## **Software Requirements Specification**

**Course Code:** INT-220

**Course Name:** SERVER SIDE SCRIPTING

**Student Names:** PIYUSH RANA, ROHIT KHANDELWAL,  
NILAX, SATADEEP KAR

**Student Registration Numbers:** 12320925, 12313816,  
12307521, 12318654

Prepared for  
Continuous Assessment 3  
Spring 2025

## Table of Contents

REVISION HISTORY .....	ERROR! BOOKMARK NOT DEFINED.
<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 PURPOSE .....	1
1.2 SCOPE .....	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .....	2
1.4 REFERENCES .....	3
1.5 OVERVIEW .....	3
<b>2. GENERAL DESCRIPTION.....</b>	<b>4</b>
2.1 PRODUCT PERSPECTIVE .....	4
2.2 PRODUCT FUNCTIONS .....	4
2.3 USER CHARACTERISTICS .....	5
2.4 GENERAL CONSTRAINTS.....	5
2.5 ASSUMPTIONS AND DEPENDENCIES .....	6
<b>3. SPECIFIC REQUIREMENTS.....</b>	<b>7</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	7
3.1.1 User Interfaces.....	7
3.1.2 Hardware Interfaces .....	7
3.1.3 Software Interfaces.....	7
3.1.4 Communications Interfaces.....	7
3.2 FUNCTIONAL REQUIREMENTS .....	7
3.2.1 <Functional Requirement or Feature #1> .....	<i>Error! Bookmark not defined.</i>
3.2.2 <Functional Requirement or Feature #2> .....	9
3.5 NON-FUNCTIONAL REQUIREMENTS .....	3
3.5.1 Performance.....	3
3.5.2 Reliability.....	3
3.5.3 Availability.....	3
3.5.4 Security .....	3
3.5.5 Maintainability.....	3
3.5.6 Portability .....	3
3.7 DESIGN CONSTRAINTS .....	3
3.9 OTHER REQUIREMENTS .....	3
<b>4. ANALYSIS MODELS.....</b>	<b>13</b>
4.1 DATA FLOW DIAGRAMS (DFD) .....	4
<b>5. GITHUB LINK.....</b>	<b>5</b>
<b>6. DEPLOYED LINK.....</b>	<b>6</b>
<b>7. CLIENT APPROVAL PROOF.....</b>	<b>7</b>
<b>8. CLIENT LOCATION PROOF.....</b>	<b>8</b>
<b>9. TRANSACTION ID PROOF.....</b>	<b>9</b>
<b>10. EMAIL ACKNOWLEDGEMENT.....</b>	<b>10</b>
<b>11. GST No.....</b>	<b>11</b>
<b>A. APPENDICES.....</b>	
A.1 APPENDIX 1.....	
A.2 APPENDIX 2.....	



# 1. Introduction

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements for the Virtual Pet Care System with Daily Reminders. The primary objective of this system is to assist pet owners in managing their pet care routines efficiently through daily reminders for tasks such as feeding, exercise, grooming, and medical checkups.

This document serves as a comprehensive guide for software engineers to understand the scope, features, and constraints of the proposed system. It ensures that all stakeholders have a clear understanding of what the system is intended to do and provides a solid foundation for system design, development, and validation.

The structure and content of this document are informed by the IEEE standards for Software Requirements Specification to ensure clarity, consistency, and completeness.

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the requirements for the Virtual Pet Care System with Daily Reminders. This system is designed to help pet owners manage and track daily care activities for their pets, such as feeding, exercise, grooming, and medical appointments, through timely notifications and reminders.

This document is intended for use by:

- Software developers and engineers who will design, implement, and test the system.
- Project managers overseeing development and deployment.
- Clients or stakeholders who want to understand the features and functionality of the system.
- Testers and QA engineers responsible for validating that the final product meets the defined requirements.

It aims to provide a clear and complete understanding of the system's intended functionality and constraints, serving as a foundation for development and future enhancements.

## 1.2 Scope

The software product to be developed is named **Virtual Pet Care System with Daily Reminders**.

This system is a web-based application designed to assist pet owners in organizing and tracking daily care activities for their pets. It will allow users to schedule and receive reminders for tasks such as feeding, walking, grooming, vaccination appointments, and other routine activities. The system will support multiple pets and enable users to customize care routines based on each pet's needs.

**The system will include the following core functionalities:**

## Project Name

- User registration and login
- Adding and managing pet profiles
- Scheduling and editing care tasks
- Sending daily automated reminders via email or notifications
- Tracking completed and pending tasks
- Dashboard for daily/weekly pet care overview

### **The system will not include:**

- Real-time health monitoring via wearable devices
- Integration with veterinary clinic databases
- Direct in-app communication with pet professionals

**Application and Objectives:** This system is intended for individual pet owners who seek a structured and convenient way to manage their pet care responsibilities. The key goals of the system include:

- Providing a centralized platform to manage multiple pets
- Ensuring timely care through customizable, user-defined reminders
- Improving consistency and efficiency in pet care routines
- Enhancing pet well-being through better care planning

This SRS aligns with the broader goal of promoting responsible pet ownership by offering a user-friendly, reliable, and accessible digital solution.

## 1.3 Definitions, Acronyms, and Abbreviations

This subsection defines the key terms, acronyms, and abbreviations used throughout this Software Requirements Specification (SRS) document to ensure clarity and consistency.

- **SRS** – Software Requirements Specification
- **UI** – User Interface
- **UX** – User Experience
- **Notification** – A message or alert sent to the user to remind them of a scheduled pet care task
- **Task Scheduler** – A system component responsible for setting and managing daily reminders for pet care activities
- **Dashboard** – The main user interface screen showing an overview of pet care tasks and statuses
- **Web App** – A web-based application accessible through a browser
- **Pet Profile** – A record in the system containing details about a specific pet (e.g., name, age, type, medical needs)
- **Reminder** – A scheduled alert that prompts the user to perform a specific action for pet care
- **Admin Panel** – A backend interface (if applicable) used to manage system data and monitor user activity

Additional technical or domain-specific terms, if any, will be defined in the Appendix section of this document.

## 1.4 References

The following documents and resources have been referenced during the preparation of this Software Requirements Specification (SRS). These materials provided guidance on software engineering best practices, user interface standards, and SRS formatting conventions.

1. **IEEE Std 830-1998** – IEEE Recommended Practice for Software Requirements Specifications
  - o *Date:* October 20, 1998
  - o *Publisher:* IEEE Computer Society
  - o *Source:* [IEEE Xplore Digital Library](#)
2. **The IEEE Guide to Software Requirements Specifications**
  - o *Publisher:* Institute of Electrical and Electronics Engineers
  - o *Used for:* Structuring and annotating the SRS document
3. **W3C Web Accessibility Guidelines (WCAG) 2.1**
  - o *Date:* June 2018
  - o *Publisher:* World Wide Web Consortium (W3C)
  - o *Source:* <https://www.w3.org/TR/WCAG21/>
  - o *Used for:* Ensuring accessibility considerations in UI design
4. **Pet Care Routine Standards – American Veterinary Medical Association (AVMA)**
  - o *Publisher:* AVMA
  - o *Used for:* Identifying common pet care tasks and timelines
  - o *Source:* <https://www.avma.org/>

Additional internal documents such as design drafts, UI mockups, or stakeholder notes are referenced in the appendices where applicable

## 1.5 Overview

This Software Requirements Specification (SRS) document is organized to provide a comprehensive understanding of the **Virtual Pet Care System with Daily Reminders**, covering both functional and non-functional aspects of the system.

The remainder of this document is structured as follows:

- **Section 2: Overall Description**  
Provides a general overview of the system, its features, user characteristics, and operating environment.
- **Section 3: Specific Requirements**  
Details all the functional and non-functional requirements of the system, including use cases, input/output specifications, and system behavior.
- **Section 4: External Interface Requirements**  
Describes the interfaces between the software and users, hardware, and other systems (if any).
- **Appendices**  
Includes any supplementary information such as glossary, diagrams, or additional references that support the main content of the SRS.

## 2. General Description

This section provides a high-level overview of the **Virtual Pet Care System with Daily Reminders**. It outlines the general factors that influence the system and its development, including user needs, system constraints, assumptions, and dependencies.

The purpose of this section is not to list specific requirements, but to offer background information that helps in understanding the rationale behind those requirements. It sets the context in which the system operates and supports a clearer interpretation of the detailed specifications presented in later sections.

Topics covered include:

- Product perspective and features
- User characteristics
- Operating environment
- Design and implementation constraints
- Assumptions and dependencies

### 2.1 Product Perspective

The **Virtual Pet Care System with Daily Reminders** is an independent, web-based application designed to help pet owners manage daily pet care routines through scheduled reminders. It is a self-contained system developed specifically for individual users and does not depend on any existing pet care platforms or third-party software.

However, the system can be extended in the future to integrate with:

- Mobile platforms for push notifications
- Veterinary clinic systems for appointment reminders
- IoT-enabled pet care devices for health and activity monitoring

This application follows a modular architecture, with separate components for user management, pet profile handling, task scheduling, and notification services. Its lightweight design ensures ease of use while offering potential for scalability and integration with external services in future versions.

### 2.2 Product Functions

The **Virtual Pet Care System with Daily Reminders** is designed to assist pet owners in organizing and managing their pet care tasks efficiently. The key functions of the system include:

- **User Registration and Authentication:** Allow users to create accounts and securely log in to access their personalized dashboard.
- **Pet Profile Management:** Enable users to add, edit, and delete pet profiles with information such as name, breed, age, medical history, and dietary needs.
- **Task Scheduling:** Allow users to schedule recurring or one-time tasks for each pet, such as feeding, walking, grooming, and vet visits.
- **Daily Reminders:** Automatically send reminders to users via notifications or email based on scheduled tasks.
- **Dashboard View:** Display a user-friendly dashboard showing daily and weekly care tasks, with status indicators for completed and pending actions.

- **Task History Tracking:** Maintain a log of completed tasks for review and record-keeping.

These core functionalities aim to promote responsible pet care by helping users maintain consistent routines and never miss important activities or appointments.

## 2.3 User Characteristics

The intended users of the **Virtual Pet Care System with Daily Reminders** are pet owners who wish to efficiently manage their pets' daily care routines. These users may vary in age, technical background, and familiarity with digital tools. Therefore, the system is designed to be intuitive, accessible, and user-friendly.

**Key user characteristics include:**

- **General Users (Pet Owners):**
  - May or may not have prior experience with similar applications
  - Require simple and clear navigation
  - Prefer mobile-friendly or responsive design for ease of use on various devices
  - Expect quick setup and easy task scheduling
- **Admin Users (Optional, if applicable):**
  - Responsible for maintaining backend data and managing user accounts
  - Require access to dashboards or system logs
  - Need more advanced controls and access to system settings

The user interface is designed with minimal complexity to cater to a non-technical audience while still being efficient for frequent and returning users.

## 2.4 General Constraints

The design and development of the **Virtual Pet Care System with Daily Reminders** are subject to several constraints that will guide the system architecture, functionality, and user experience.

These constraints include:

- **Technology Constraints:**

The system is developed as a web-based application using JavaScript (React) for the frontend and a Node.js backend. The system must be compatible with modern web browsers (Chrome, Firefox, Edge, Safari) and should function optimally on both desktop and mobile devices.
- **User Interface Design:**

The system must be designed with simplicity in mind, as the target users may not be highly tech-savvy. The user interface must be accessible, with minimal technical jargon, and should comply with basic web accessibility standards (e.g., WCAG 2.1).
- **Data Storage:**

User and pet data must be stored securely, complying with data protection laws such as GDPR or equivalent local regulations. All data must be stored in an encrypted database, with user authentication and authorization protocols in place.
- **Performance Constraints:**

The application should respond quickly to user inputs, especially for tasks like adding pet profiles, scheduling reminders, and receiving notifications. The system should aim for load times under 3 seconds for the main features.



- **Third-Party Integrations:**  
Initially, the system will not integrate with external services like veterinary clinics or IoT pet devices. Future scalability must account for these integrations.
- **Budget and Time Constraints:**  
The project must be completed within a specified budget and timeline, limiting the scope of additional features and third-party dependencies in the initial release.

## 2.5 Assumptions and Dependencies

This subsection outlines the key assumptions made during the development of the **Virtual Pet Care System with Daily Reminders** and any dependencies that may affect the system's requirements. These factors are not constraints on design, but changes to them could impact the system's functionality or the requirements as described in this document.

### Key Assumptions:

- **Availability of Internet Access:**  
It is assumed that users will have reliable internet access to use the web-based application. If users experience limited or no internet access, the system may not function as intended.
- **Browser Compatibility:**  
It is assumed that users will be using modern, updated browsers such as Google Chrome, Firefox, Safari, or Edge. The system may not be fully compatible with outdated browsers.
- **User Devices:**  
The system assumes that users will access the platform through a desktop or mobile device that meets certain minimum technical specifications (e.g., screen size, resolution).
- **User Technical Literacy:**  
It is assumed that the target audience (pet owners) will have a basic understanding of using web-based applications, including setting up reminders and managing user accounts. This assumption may need to be adjusted based on user feedback.

### Dependencies:

- **Third-Party Notification Service:**  
The system depends on external services for sending email and push notifications to users. Any limitations or outages of these services could impact the timely delivery of reminders.
- **Data Storage Provider:**  
The system depends on the cloud service provider for secure data storage and backup. Any changes in service terms, outages, or access restrictions could impact the availability of the system.
- **Operating System:**  
The application is assumed to be deployed on a cloud platform that supports standard web technologies. If the hosting environment changes (e.g., operating system or server software), adjustments to the system may be required.

### 3. Specific Requirements

This section outlines the detailed functional and non-functional requirements for the **Virtual Pet Care System with Daily Reminders**. These requirements serve as the foundation for the system's design, development, and testing.

Each requirement in this section is carefully defined to ensure that it is:

- **Correct:** The requirements must accurately reflect the needs of the system as defined by the stakeholders.
- **Traceable:** Each requirement will be linked to its origin and will also trace forward to design, implementation, and testing phases.
- **Unambiguous:** Each requirement is clearly stated without room for misinterpretation.
- **Verifiable:** Requirements must be testable, ensuring that each can be validated through appropriate testing methods.
- **Prioritized:** Requirements will be ranked by importance and stability to ensure the most critical features are developed and tested first.
- **Complete:** All necessary functionalities and conditions are captured, ensuring no aspects are overlooked.
- **Consistent:** No conflicting or contradictory requirements will be present.
- **Uniquely identifiable:** Each requirement will be uniquely numbered for easy reference throughout the project lifecycle (e.g., 3.1, 3.2.1).

While this section focuses on requirements, it is important to note that it is not the software design document. The goal is to outline the necessary functional and non-functional requirements without prematurely specifying design constraints. These requirements provide a clear blueprint for the development team to follow, ensuring the software meets the expectations and needs of the users.

#### 3.1 External Interface Requirements

#### 3.2 Functional Requirements

This section provides a detailed description of the specific features and functionalities of the **Virtual Pet Care System with Daily Reminders**. The requirements are organized to define the system's core capabilities, ensuring that each feature meets the needs of the users. Some requirements may be expressed in the use-case format to illustrate specific scenarios and interactions.

Key functional requirements include:

- **User Registration and Login:**  
Users must be able to create an account, log in securely, and reset their passwords if necessary. This functionality ensures that only authorized users can access their pet care data.
- **Pet Profile Management:**  
Users must be able to create, view, edit, and delete pet profiles. A pet profile includes details like the pet's name, age, breed, medical information, and dietary needs.
- **Task Scheduling:**  
Users should be able to schedule daily or recurring reminders for pet care tasks, such as

feeding, walking, grooming, or medical checkups. Each task should be customizable (e.g., frequency, time, and type).

- **Reminder Notifications:**

The system should send notifications (via email or mobile alerts) to users based on their scheduled tasks. These reminders should be delivered in a timely manner and include relevant details such as the type of task, the pet's name, and the scheduled time.

- **Dashboard Overview:**

The system must provide a dashboard displaying the user's scheduled tasks for the day, with the ability to mark tasks as completed. The dashboard should also show pending tasks, past activities, and upcoming reminders.

- **History and Record-Keeping:**

Users should be able to view a history of completed tasks for each pet. This feature allows for easy tracking of past care routines and any changes to the schedule.

### **3.2.1 User Registration and Login**

#### **3.2.1.1 Introduction**

**This feature allows users to register, log in, and securely access their personalized pet care dashboard. It ensures that only authorized users can access and manage their pet care tasks and profiles.**

#### **3.2.1.2 Inputs**

- **User Information:**
  - **Username**
  - **Email Address**
  - **Password**
  - **(Optional) Phone number (for recovery purposes)**
- **Registration:**
  - **Users must enter valid information to create a new account.**
  - **The system will require confirmation of the email address via a verification code.**

#### **3.2.1.3 Processing**

- **Account Creation:**

**The system verifies the uniqueness of the username and email. If the credentials are valid, a new user account is created in the system.**
- **Login Validation:**

**Upon login, the system checks the entered credentials (username/email and password) against stored values. If the credentials match, access is granted.**

  - **If incorrect, the user is prompted with an error message.**

#### **3.2.1.4 Outputs**

- **Successful Registration:**

**A confirmation message is sent to the user's email with a verification code to activate the account.**
- **Successful Login:**

**A dashboard is presented with the user's personalized pet care tasks and settings.**
- **Error Messages:**
  - **"Username/Email already exists" (during registration)**
  - **"Invalid username/email or password" (during login)**
  - **"Account not verified" (if email verification is pending)**

#### **3.2.1.5 Error Handling**

- **Invalid Inputs:**

**If the user inputs invalid or incomplete data during registration, an error message is**

displayed with guidance on what needs to be corrected (e.g., "Email format is invalid").

- **Failed Login Attempts:**  
After a specified number of failed login attempts, the user account will be temporarily locked for security purposes, with a notification on how to recover the account.

...

### 3.5 Non-Functional Requirements

The following non-functional requirements outline the system's performance, reliability, availability, security, maintainability, and portability. These attributes are critical for ensuring that the **Virtual Pet Care System with Daily Reminders** operates efficiently and meets the expectations of its users.

#### 3.5.1 Performance

- **Response Time:**  
95% of user interactions (e.g., adding a pet profile, scheduling a task, and viewing the dashboard) should be processed in under 2 seconds.
- **Throughput:**  
The system should be capable of handling at least 100 concurrent users without significant degradation in performance.
- **Scalability:**  
The system should be able to scale to accommodate 10,000 active users without a noticeable decrease in performance.

#### 3.5.2 Reliability

- **System Uptime:**  
The system should have an uptime of 99.9% per month, excluding scheduled maintenance periods.
- **Mean Time Between Failures (MTBF):**  
The system should have a minimum MTBF value of 30 days, ensuring reliability and stability.
- **Error Recovery:**  
The system must automatically recover from unexpected failures (e.g., server crashes) within 5 minutes with minimal impact on the user experience.

#### 3.5.3 Availability

- **Service Availability:**  
The system must be available 24/7, with no more than 1 hour of planned downtime per month for maintenance.
- **Geographical Accessibility:**  
The system should be accessible from any region, provided the user has a stable internet connection.

#### 3.5.4 Security

- **Data Protection:**  
All sensitive data (e.g., user credentials, pet health data) must be encrypted using at least AES-256 encryption.

- **Authentication:**  
The system should require multi-factor authentication (MFA) for users with administrative access or sensitive data.
- **Authorization:**  
Only authorized users should have access to specific pet profiles and care tasks. Role-based access control (RBAC) should be implemented.
- **Data Privacy:**  
The system must comply with GDPR and other relevant privacy regulations, ensuring that user data is protected and not shared with third parties without explicit consent.

### 3.5.5 Maintainability

- **Code Quality:**  
The system's codebase should adhere to established coding standards (e.g., clean code principles) to ensure maintainability.
- **Update Frequency:**  
System updates (bug fixes, feature improvements) should be rolled out at least once every two weeks.
- **Logging and Monitoring:**  
The system must have comprehensive logging and monitoring tools in place to detect and alert for potential issues.

### 3.5.6 Portability

- **Cross-Browser Compatibility:**  
The system must function correctly on major web browsers (Chrome, Firefox, Safari, Edge) with no degradation in user experience.
- **Mobile Responsiveness:**  
The system should be fully responsive, providing an optimal user experience across both mobile devices (smartphones and tablets) and desktops.
- **Deployment Flexibility:**  
The system should be deployable on any cloud platform (e.g., AWS, Google Cloud, Azure) without requiring significant changes to the codebase.

## 3.7 Design Constraints

This section outlines the design constraints that will impact the development of the **Virtual Pet Care System with Daily Reminders**. These constraints arise from various standards, company policies, hardware limitations, and other external factors that must be considered during the system's design and development.

### 3.6.1 Standards and Regulations

- **Data Privacy Compliance:**  
The system must comply with data privacy laws such as GDPR (General Data Protection Regulation) and HIPAA (Health Insurance Portability and Accountability Act) to ensure the secure handling of personal and health-related data of pets.
- **Accessibility Standards:**  
The system must meet WCAG 2.1 (Web Content Accessibility Guidelines) to ensure accessibility for users with disabilities, including screen reader support and keyboard navigation.

### 3.6.2 Company Policies

- **Branding Guidelines:**  
The system's user interface (UI) design must adhere to the company's branding guidelines, including color schemes, fonts, and logo placement, ensuring consistency with other company products and services.
- **Security Protocols:**  
All security protocols, including encryption standards for user data and authentication processes, must align with the company's internal security policies to ensure that the system is secure and protected from unauthorized access.

### 3.6.3 Hardware Limitations

- **Device Compatibility:**  
The system is designed to be compatible with common user devices, including smartphones, tablets, and desktops. It must be optimized for devices with a minimum screen size of 320px (mobile) and should function across modern web browsers (e.g., Chrome, Firefox, Safari, Edge).
- **Cloud Hosting Requirements:**  
The system will be hosted on cloud infrastructure, and the chosen cloud platform (e.g., AWS, Google Cloud) must support necessary services like database hosting, notifications, and scalable storage.

### 3.6.4 Third-Party Services

- **Email Notification Services:**  
The system will rely on third-party services for sending email notifications to users for task reminders. These services must meet the company's performance and reliability standards.
- **Push Notification Services:**  
The system will use a third-party push notification service (e.g., Firebase Cloud Messaging) to send reminders. The service must ensure timely delivery and allow customization for user preferences.

## 3.9 Other Requirements

This section captures any additional requirements that are not covered in the previous sections but are important for the successful implementation and operation of the **Virtual Pet Care System with Daily Reminders**.

### 3.7.1 User Preferences and Customization

- **Customizable Reminder Notifications:**  
Users should have the ability to customize the frequency, time, and method (email or push notification) of pet care reminders. This feature will ensure that users can tailor the system to fit their individual schedules and preferences.

### 3.7.2 Integration with External Systems

- **Integration with Calendar Services:**  
The system should provide an option for users to sync pet care tasks with their personal calendar (Google Calendar, iCal, etc.) to help users manage their schedules more effectively.

### 3.7.3 Multilingual Support

- **Localization:**  
The system should support multiple languages to cater to users from different linguistic

backgrounds. This can be achieved by providing language preferences in the user settings, with initial support for English and Spanish.

### **3.7.4 System Logging and Monitoring**

- **Error Logging:**

The system must have a built-in logging mechanism to capture errors, user activity, and system performance data. Logs should be stored securely and available to the development team for troubleshooting and performance optimization.

### **3.7.5 Backup and Recovery**

- **Data Backup:**

The system must perform daily backups of user data (pet profiles, task schedules, etc.) to ensure that data is recoverable in case of a system failure. Backups should be stored securely and encrypted.

## **5. Analysis Models**

This section lists the analysis models used to develop the specific requirements outlined in the previous sections of the **Virtual Pet Care System with Daily Reminders SRS**. Each model is provided with an introduction, a narrative description, and a traceability link to the corresponding requirements in the SRS.

### **3.8.1 Use Case Diagrams**

- **Introduction:**

Use case diagrams visually represent the functional interactions between the system and its users. They provide a clear overview of how users interact with the system to accomplish various tasks such as creating pet profiles, scheduling reminders, and viewing pet care history.

- **Narrative Description:**

The use case diagram includes actors such as "User," "Admin," and "System" to represent interactions such as registering a user, logging in, creating a pet profile, and setting reminders. Each interaction is linked to a specific functional requirement.

- **Traceability:**

- **Use Case 1: Register User → Functional Requirement 3.2.1: User Registration**
- **Use Case 2: Schedule Task → Functional Requirement 3.2.2: Task Scheduling**
- **Use Case 3: Send Reminder → Functional Requirement 3.2.3: Reminder Notifications**

### **3.8.2 Data Flow Diagrams (DFD)**

- **Introduction:**

Data Flow Diagrams illustrate how information flows through the system. They map out how data is input, processed, and output by the system, ensuring that all components of the system are aligned with the defined requirements.

- **Narrative Description:**

The DFD provides a detailed view of the interactions between the user interface, the database, and the notification system. For example, user data flows from the registration form to the database, and task information flows to the reminder notification system.

- **Traceability:**



- **DFD Process 1: User Registration** → **Functional Requirement 3.2.1: User Registration**
- **DFD Process 2: Schedule Task** → **Functional Requirement 3.2.2: Task Scheduling**
- **DFD Process 3: Notification System** → **Functional Requirement 3.2.3: Reminder Notifications**

### **3.8.3 Entity-Relationship Diagram (ERD)**

- **Introduction:**

The Entity-Relationship Diagram outlines the relationships between different entities in the system, such as users, pets, and tasks. This model ensures that the system's database is designed to accommodate all necessary data structures.

- **Narrative Description:**

The ERD depicts entities like "User," "Pet," "Task," and "Reminder," with relationships indicating that a user can have multiple pets, each pet can have multiple scheduled tasks, and each task can have associated reminders.

- **Traceability:**

- **Entity "User"** → **Functional Requirement 3.2.1: User Registration**
- **Entity "Pet"** → **Functional Requirement 3.2.2: Pet Profile Management**
- **Entity "Task"** → **Functional Requirement 3.2.2: Task Scheduling**

## **4.1 Data Flow Diagrams (DFD)**

This section provides a visual representation of the data flow within the **Virtual Pet Care System with Daily Reminders**. The Data Flow Diagram (DFD) illustrates how data is input, processed, stored, and output in the system. It provides a clear overview of the system's components and their interactions.

### **4.1.1 Context Diagram**

The **Context Diagram** represents the system as a single process and shows how it interacts with external entities such as users, email services, and cloud storage. It highlights the flow of data between the system and these external entities.

- **External Entities:**

- **Users:** Input pet profiles, schedules tasks, and receive reminders.
- **Email Service:** Sends email reminders to users.
- **Push Notification Service:** Sends push notifications to users.

### **4.1.2 Level 1 DFD**

The **Level 1 DFD** breaks down the main system process into its sub-processes, showing the flow of data between these processes and data stores.

- **Processes:**

- **User Registration and Login:** Handles user credentials and session management.
- **Pet Profile Management:** Stores and retrieves information about pets.
- **Task Scheduling:** Allows users to schedule and modify pet care tasks.
- **Reminder Notification:** Sends notifications based on the scheduled tasks.

- **Data Stores:**

- **User Database:** Stores user information (login credentials, settings).
- **Pet Database:** Stores pet details (name, age, medical information).
- **Task Database:** Stores pet care task schedules and reminders.

## A. Appendices

Appendices are provided to offer additional helpful information that supplements the main body of the **Virtual Pet Care System with Daily Reminders** Software Requirement Specification (SRS). These appendices contain materials that may assist stakeholders in understanding the system's scope, design, or other related elements.

### A.1 Conceptual Documents

This appendix contains initial conceptual documents for the system, including sketches, wireframes, and high-level descriptions of system features. These documents provide a preliminary visual and functional overview of the Virtual Pet Care System.

### A.2 User Personas

This appendix includes detailed descriptions of the target users of the system, such as pet owners, veterinary professionals, and system administrators. These personas are used to guide the development process and ensure the system meets the needs of all user types.

### A.3 System Architecture Overview

This appendix provides an overview of the system architecture, including the technologies used (e.g., cloud hosting, databases, notification systems) and how different components interact with each other.

### A.4 Meeting Minutes

This appendix contains records of meetings with customers and stakeholders, providing insights into the requirements gathering process, feature prioritization, and any decisions made during discussions. These documents serve as a reference for understanding the rationale behind certain design and feature choices.

### A.5 Glossary of Terms

This appendix provides definitions of any technical terms, acronyms, and abbreviations used throughout the SRS. It ensures that all stakeholders have a clear understanding of the terminology employed in the document.