**SOFTWARE REQUIREMENTS SPECIFICATION**

**FOR**

**PET ADOPTION SYSTEM**

**PREPARED BY: -**

*JAYA GHAYATHIRI B S*

*KAMALI V M*

*KOUSHIKA T*

**1.Introduction**

**1.1 Purpose:**

The purpose of a pet adoption system is to facilitate the responsible placement of animals in suitable homes. It aims to connect adoptable pets with individuals or families looking to provide them a loving environment. This system helps reduce the number of stray animals, promotes responsible pet ownership, and ensures animals find caring homes.

**1.2 Document Conventions**

* Entire document should be justified.
* Convention for Main title
  + Font face: Times New Roman
  + Font style: Bold
  + Font Size: 14
* Convention for Sub title
* Font face: Times New Roman
* Font style: Bold
* Font Size: 12
* Convention for body
* Font face: Times New Roman
* Font style: Bold
* Font Size: 12

**1.3 Scope of Development Project**

* The scope of a Pet Adoption System is extensive, as it addresses various aspects related to facilitating the adoption process of pets and ensuring a seamless experience for both adopters and administrators. This comprehensive system aims to leverage technology to promote responsible pet ownership, connect potential adopters with available animals, and streamline the entire adoption journey.
* A pet adoption system refers to a digital platform or software designed to facilitate the adoption of pets. The scope of a pet adoption system development project encompasses the planning, features, and functionalities to be included in the system.
* A pet adoption system involves creating a user-friendly platform connecting adopters with pets in need. Key elements include intuitive interfaces, detailed pet profiles, advanced search and matching algorithms, social media integration, streamlined adoption processes, veterinary record integration, fundraising features, feedback mechanisms, mobile compatibility, legal compliance, educational resources, and continuous improvement through updates and user feedback. The system aims to simplify and enhance the pet adoption experience while promoting responsible ownership.
* Develop an intuitive and user-friendly interface for both potential adopters and organisations managing the adoption process. Include features such as easy navigation, search filters, and a streamlined adoption process.
* Allow users to share pet profiles on social media platforms and enable social media logins for user convenience.

**1.4 Definitions, Acronyms and Abbreviations**

JAVA -> platform independence

SQL-> Structured query Language

ER-> Entity Relationship

UML -> Unified Modelling Language

IDE-> Integrated Development Environment

SRS-> Software Requirement Specification

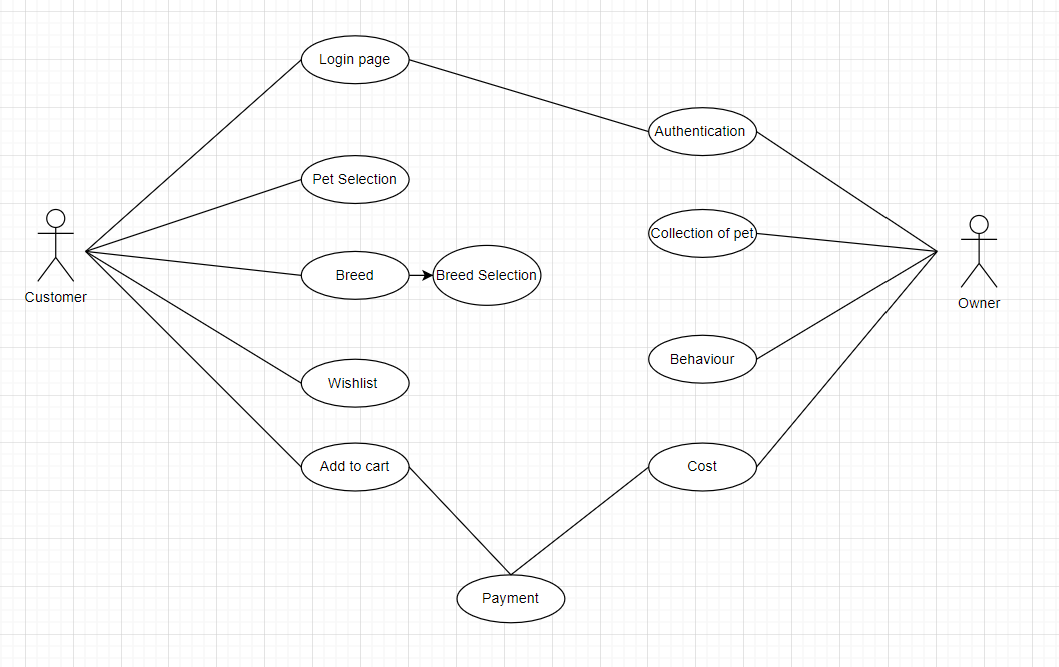
**1.5 References**

* Books
* Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson
* Software Requirements (Microsoft) Second Edition by Karl E. Wiegers.
* Websites
* [**http://www.slideshare.net/**](http://www.slideshare.net/)
* [**http://ebookily.net/doc/srs-library-management-system**](http://ebookily.net/doc/srs-library-management-system)

**2. Overall Descriptions**

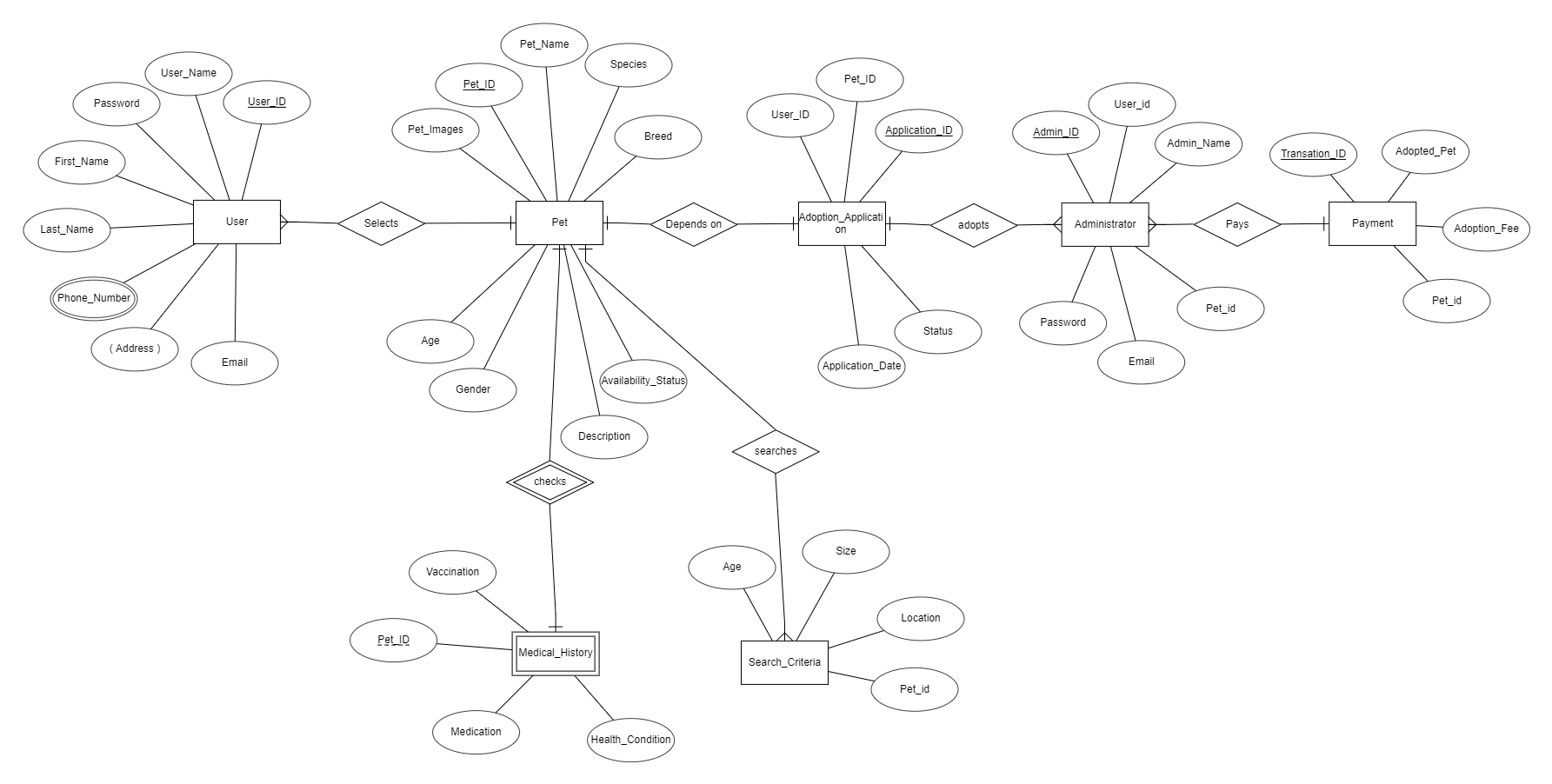
**2.1 Product Perspective**

Use Case Diagram of Library Management System.



**2.2 Product Function**

Entity Relationship Diagram of Library Management System.



**2.3 User Classes and Characteristics**

The pet adoption system accommodates various user roles, primarily categorized as Adopters and Administrators. Adopters may include individuals or families seeking to adopt a pet, while Administrators manage the overall system.

**Adopters:**

**Viewers/Prospective Adopters:**

* Can browse available pets.
* View different categories and breeds of pets.
* Access detailed profiles of each pet, including photos and information.

**Registered Adopters:**

* Own an account in the adoption system.
* View pets they have adopted or shown interest in.
* Put in adoption requests for specific pets.
* View the history of adopted pets.

**Administrators:**

* Pet Organizations/Shelters:
* Manage and update pet profiles.
* Review and process adoption applications.
* Schedule meet-and-greet sessions.
* Monitor and update pet health records.

**Pet Organizations/Shelters:**

* Manage and update pet profiles.
* Review and process adoption applications.
* Schedule meet-and-greet sessions.
* Monitor and update pet health records.

**2.4 Operating Environment**

Online Platform: A web-based or mobile application platform where users can browse available pets, submit adoption applications, and access information about the adoption process

Payment Gateway: If applicable, a secure payment system for adoption fees, donations, or any financial transactions associated with the adoption process.

Integration with Veterinary Records: If available, a system to access and manage the health records of animals, ensuring transparency about the pet's medical history.

**2.5 Assumptions and Dependencies**

The assumptions are: -

* Users and adoption agencies have reliable internet access to utilize the online platform for browsing, applying, and managing the adoption process.
* Users are familiar with basic online navigation and are comfortable using web or mobile applications.
* The system assumes that the adoption agency and users comply with local and national laws and regulations related to pet adoption.
* Information provided about the health and behavior of pets is assumed to be accurate, and regular veterinary assessments are conducted.
* Adoption agencies assume they have the necessary resources to care for animals, including shelter, veterinary care, and staffing.

The dependencies are: -

* + - The system depends on accurate and up-to-date data regarding available pets, adopter information, and the status of the adoption process.
    - Integration with third-party services, such as payment gateways for adoption fees or external databases for additional pet information, may create dependencies.
    - The availability and reliability of the servers hosting the pet adoption system are crucial for uninterrupted service.
    - Adoption agency staff must be trained to use the system effectively, ensuring proper management of the adoption process and accurate data input.
    - The system depends on robust security measures to protect user data, financial transactions, and other sensitive information.
    - Dependencies on adherence to legal and regulatory requirements mean the system must adapt to any changes in these regulations.

**2.6 Requirement**

**Software Configuration: -**

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP

Language: Java Runtime Environment, Net beans 7.0.1 (front end)

Database: MS SQL Server (back end)

**2.7 Data Requirement**

A robust pet adoption system relies on crucial data elements to facilitate responsible and informed placements. Pet information, including name, breed, medical history, and behavior traits, enables prospective adopters to make well-informed decisions. Adopter details, such as contact information and living situation, aid in assessing suitability. Application data, encompassing reasons for adoption and references, contributes to the thorough vetting of potential adopters. Rich media content, like photos and videos, enhances the visual understanding of available pets.

Legal documentation, including adoption contracts and fee records, ensures transparency and compliance. Communication logs and geographical data support effective interaction and logistics. Feedback from adopters provides insights into the adoption experience, while staff information and financial data contribute to internal management and sustainability. Prioritizing data security and privacy is paramount in creating a trustworthy and efficient pet adoption system.

**3. External Interface Requirement**

**3.1 GUI**

**Homepage:**

* Welcoming interface with high-quality images of pets available for adoption.
* Clear navigation links to different sections: Browse Pets, Adoption Process, About Us, Contact, etc.
* Featured pets’ section to highlight animals in need of adoption.

**Browse Pets:**

* Grid or list view of available pets with thumbnail images and basic details (name, breed, age).
* Filter options based on criteria such as species, breed, age, and location.
* Search bar for users to find specific pets or breeds.

**Pet Details Page:**

* Larger images of the selected pet with a detailed description.
* Information about the pet's health, behaviour, and any special needs.
* Adoption fee details and a button to initiate the adoption process.

**Adoption Process:**

* Step-by-step guide explaining the adoption process.
* Online application form with fields for adopter details, living situation, and preferences.
* Clear instructions and guidance on required documentation and fees.

**User Account:**

* Profile dashboard displaying the user's adoption history, application status, and saved favourites.
* Editable profile information and account settings.
* Notifications for application status updates.

**Admin Dashboard:**

* Overview of adoption statistics, pending applications, and recent activity.
* Tools for managing pet listings, user accounts, and applications.
* Communication features to contact adopters and update pet information.

**4. System Features**

User Interface:

* Intuitive Design: Ensure a user-friendly and intuitive interface to facilitate easy navigation and use for potential adopters, shelter staff, and administrators.
* Accessibility: Design the system to be accessible to users with disabilities, following accessibility standards such as WCAG.

User Management:

* User Registration and Profiles: Allow users to create accounts, complete profiles, and manage their adoption preferences.
* Authentication: Implement secure authentication methods, including password protection and optional multi-factor authentication.

Pet Listings:

* Comprehensive Pet Profiles: Include detailed information about each pet, such as age, breed, health status, behaviour, and photos.
* Search and Filter Options: Provide robust search and filtering capabilities for users to easily find pets based on their preferences.

Adoption Process:

* Application Forms: Enable users to submit adoption applications online.
* Approval Workflow: Implement a workflow for reviewing and approving adoption applications.
* Communication: Facilitate communication between adopters and shelter staff through the platform.

**5. Other Non-Functional Requirements**

**5.1 Performance Requirement**

The system should be able to handle a scalable number of users and pets without a significant degradation in performance. The system should support a minimum of X concurrent users during peak times. Ensure that the system is responsive on various devices, including smartphones and tablets, with an acceptable page load time. The system should support a minimum of X transactions per second.

**5.2 Safety Requirement**

All pets listed in the system should have a valid health certification from a licensed veterinarian. Provide adoption counselling services to potential adopters. Ensure that all adopted pets are microchipped for identification purposes. Conduct behavioural assessments for pets before listing them for adoption. Establish and communicate emergency procedures for adopters, including contact information for veterinary services and emergency shelters.

**5.3 Security Requirement**

User Authentication: Implement strong authentication mechanisms such as multi-factor authentication (MFA) to ensure that only authorized individuals can access the system.

**Data Protection:**

Encryption: Use encryption techniques (both in transit and at rest) to protect sensitive data such as user credentials, personal information, and financial transactions.

Pseudonymization and Anonymization: If possible, use techniques like pseudonymization and anonymization to protect the identities of users and pets.

**Secure Communication:**

Secure Protocols: Employ secure communication protocols (e.g., HTTPS) to protect data as it travels between clients and the server.

**Input Validation:**

Data Validation: Implement input validation mechanisms to prevent common vulnerabilities such as SQL injection and cross-site scripting (XSS).

**5.4 Requirement attributes**

* There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
* The project should be open source
* The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
* The user be able to easily download and install the system

**5.5 Business Rules**

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

**5.6 User Requirement**

The users of the system are members and Librarian of the university who act as administrator to maintain the system. The members are assumed to have basic knowledge of the computers and internet browsing. The administrators of the system should have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

The admin provides certain facilities to the users in the form of: -

* + - Backup and Recovery
    - Forgot Password
    - Data migration i.e. whenever user registers for the first time then the data is stored in the server
    - Data replication i.e. if the data is lost in one branch, it is still stored with the server
    - Auto Recovery i.e., frequently auto saving the information
    - Maintaining files i.e., File Organization
    - The server must be maintained regularly and it has to be updated from time to time

**6. Other Requirements**

**6.1 Data and Category Requirement**

The pet adoption system requires detailed data for pet profiles (name, breed, age, size, images, traits) and user accounts (unique IDs, passwords, contact details, adoption history). Administrators need access credentials and activity logs. Categories include pet types (dogs, cats), user roles (adopters, administrators), and event types (adoption events, return reminders). These components ensure effective management and organization within the pet adoption system.

**6.2 Appendix**

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance, Perspective, Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

**6.3 Glossary**

The following are the list of conventions and acronyms used in this document and the project as well:

* Administrator: A login id representing a user with user administration privileges to the software
* User: A general login id assigned to most users
* Client: Intended users for the software
* SQL: Structured Query Language; used to retrieve information from a database
* SQL Server: A server used to store data in an organized format
* Layer: Represents a section of the project
* User Interface Layer: The section of the assignment referring to what the user interacts with directly
* Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
* Data Storage Layer: The section of the assignment referring to where all data is recorded
* Use Case: A broad level diagram of the project showing a basic overview
* Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes
* Interface: Something used to communicate across different mediums
* Unique Key: Used to differentiate entries in a database

**6.4 Class Diagram**

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e., objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization.

