

Project Name:
Pet Adoption Management System (PAMS)

Course Title: Software Engineering
Course Code: CSE 305



Submitted To

Pankaj Bhowmik

Lecturer

Department of Computer Science & Engineering

Submitted By

MD. Tanjimul Haque Meet

Student ID: 2002049

Level: 3, Semester: I

Department of Computer Science & Engineering

**Hajee Mohammad Danesh Science and Technology University,
Dinajpur-5200**

Table of contents

➤ Introduction.....	3
➤ Overview of SDLC Phases.....	4
➤ Planning.....	5
➤ Defining	6
➤ Designing.....	7
➤ Building.....	8
➤ Testing.....	9
➤ Deployment and Maintenance.....	10
➤ Conclusion.....	11

Introduction

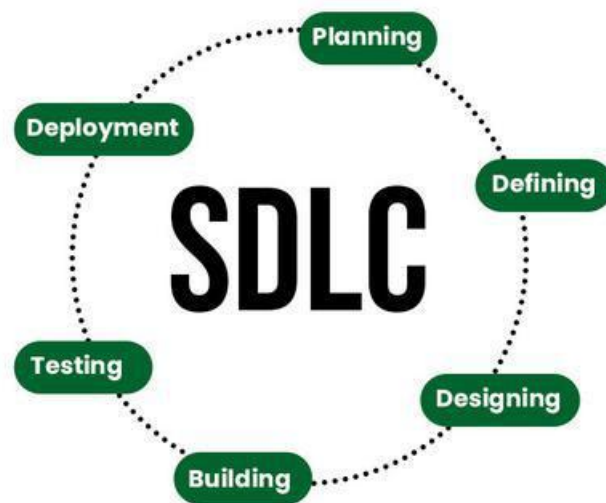
Pet adoption is a critical part of animal welfare, offering abandoned and stray animals a chance at a loving home. A **Pet Adoption Management System (PAMS)** is an automated platform that facilitates the process of pet adoption, helping both potential pet owners and animal shelters manage the adoption process.

This report explores how the Software Development Life Cycle (SDLC) can be applied to develop a robust, user-friendly Pet Adoption Management System, ensuring the delivery of high-quality software that meets stakeholder needs.

Overview of SDLC Phases

The Software Development Life Cycle (SDLC) is a structured approach to software development that involves several key stages to ensure the project is executed methodically and delivers a high-quality product. The SDLC process typically consists of the following six phases:

1. **Planning**
2. **Defining**
3. **Designing**
4. **Building**
5. **Testing**
6. **Deployment**



Each phase serves a distinct purpose, ensuring that the final product is functional, efficient, and of high quality. Below, we apply the SDLC methodology to the development of PAMS.

1. Planning

The Planning phase is the foundation for the entire Pet Adoption Management System (PAMS). This phase involves identifying the scope, goals, resources, and budget for the project. The key objective is to define the problem and identify how the system will address the needs of the users, which in this case include pet adoption agencies, prospective pet owners, and administrators.

Activities:

- **Requirement Gathering:** Engage with stakeholders, such as pet shelters, animal adoption agencies, and potential adopters, to gather and document detailed requirements.
- **Feasibility Study:** Analyze the technical, operational, and financial feasibility of the project to ensure it is realistic and achievable.
- **Defining Scope:** Define the features and functionalities that will be included in the PAMS, such as pet profiles, adoption application forms, user authentication, and search filters for pet adoption.
- **Resource Allocation:** Assign teams for development, testing, and deployment phases, and estimate the required tools and technologies.
- **Risk Assessment:** Identify potential risks like delays in development or technology limitations and establish mitigation strategies.

2. Defining

In the Defining phase, the project's requirements are clarified in detail. This includes defining the system's functional and non-functional requirements, ensuring that both business and technical perspectives are covered.

- **Functional Requirements:** Define specific features and functionality, such as user registration, pet profile creation, search and filter options, adoption application, and communication channels.
- **Non-Functional Requirements:** Focus on performance, security, scalability, and usability of the PAMS.
- **User Stories & Use Cases:** Develop user stories and use case diagrams to show how different types of users (adopter, shelter administrator, etc.) will interact with the system.
- **System Architecture:** Outline the high-level system architecture, including how the components will interact (front-end, back-end, database, etc.).
- **Data Modeling:** Design data models to represent pet information, user profiles, adoption records, and system interactions.

3. Designing

The design phase involves creating the system's architecture, database structure, and user interface.

■ **System Architecture:**

- **Client-Server Architecture:** The system can be implemented using a client-server architecture where the client is a web-based or mobile application that interacts with the server, which handles business logic, data management, and user authentication.
- **Modular Design:** The system should have separate modules for user management, pet management, and adoption processing.

■ **Database Design:**

- Design the database schema to handle various entities like Pets, Adopters, Adoption Applications, Shelters, and Users.
- **Entity-Relationship (ER) Diagram:** The ER diagram will help model the relationships between the different entities. For instance, the Pets table will be linked to the Shelters table, and adopters will be linked to their adoption applications.

■ **User Interface Design:**

- Develop wireframes and prototypes to illustrate how users will interact with the system. For instance, pet profile pages will include details such as pet breed, age, size, and adoption status, while adoption application forms will include fields for personal information and pet preferences.

• **API Design:**

- Define RESTful APIs to handle interactions between the front end (client) and back end (server), such as adding new pets, submitting adoption requests, and retrieving pet listings.

4. Building

In the building phase, the actual development of the system takes place. This includes coding and implementing the features and functionalities as outlined in the design phase.

- **Technology Stack Selection:** Choose appropriate technologies for system development:
 - Front-end: HTML, CSS, JavaScript, React for building user-friendly interfaces.
 - Back-end: Node.js with Express for managing API requests.
 - Database: MySQL for data storage.
 - Authentication: JWT (JSON Web Tokens) for secure user authentication.

- **Feature Implementation:** Code the key features of PAMS, such as:
 - Pet listing and management (adding/editing/removing pets).
 - Adoption application forms and submission processes.
 - Search functionality for filtering pets based on breed, age, etc.
 - Notification system for updates and approvals.

- **Version Control:** Use Git for code management and collaboration among developers.

5. Testing

The testing phase ensures that the system works as expected, meets user requirements, and is free of defects. It helps identify bugs or performance issues that need to be addressed before deployment.

- **Unit Testing:** Test individual components to ensure they work correctly.
- **Integration Testing:** Verify that all components of the system, including the front-end, back-end, and database, integrate seamlessly.
- **User Acceptance Testing (UAT):** Involve actual users in testing the system to ensure it meets their needs and expectations.
- **Performance Testing:** Test the system under heavy load to ensure it can handle large numbers of users and pets without performance degradation.
- **Security Testing:** Ensure the system is secure, protecting personal data and preventing vulnerabilities like SQL injection and cross-site scripting.

6. Deployment & Maintenance

The deployment phase involves releasing the system into a production environment where it can be used by end-users. It also includes providing support and training to users.

- **Production Deployment:** Deploy the application to a cloud platform, ensuring that it is live and accessible to users.
- **User Training:** Provide training materials or sessions for shelter staff on managing pet listings and adoption applications and for adopters on using the system.
- **Documentation:** Deliver comprehensive documentation, including system operation guides, FAQs, and API references for developers and users.
- **Monitoring:** Implement monitoring systems to track the system's performance and usage, ensuring it runs smoothly post-deployment.

The maintenance phase ensures that the system continues to function smoothly after deployment. It includes bug fixes, updates, and regular performance checks.

- **Bug Fixes:** Address any issues or bugs reported by users after the system is live.
- **System Updates:** Regular updates to add new features or improve existing ones, such as incorporating more filters for pet search or adding a messaging system for adopters to communicate with shelters.
- **Performance Monitoring:** Continuously monitor the system's performance and make adjustments as necessary to ensure it can handle increasing data and user traffic.

Conclusion

The **Pet Adoption Management System (PAMS)** is a prime example of how the **SDLC** can be applied to create a reliable, user-friendly software system. Through the phases of **Planning, Defining, Designing, Building, Testing, and Deployment**, the project can ensure that PAMS meets the needs of its stakeholders, functions efficiently, and remains reliable and scalable. By following these SDLC steps, PAMS will provide a seamless experience for pet shelters, adopters, and administrators, ultimately helping to match animals with loving homes.