Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2025), B.Sc. in CSE (Day)

PET Management System

Course Title: Integrated Design Project II
Course Code : CSE-406
Section : 213 D7

Students Details

Name	ID
MD Dulal Hossain	213902116
MD Rabby Khan	213902037
Mostak Ahammed	213902126

Submission Date: 11-04-2025

Course Teacher's Name: Sharifur Rahman
Designation: Lecturer
Department of CSE , GUB .

[For teachers use only: Don't write anything inside this box]

Lab Project Status

Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Pet Management System Implementation Steps and Justification

1.1 System Implementation Steps for Pet Management System

The implementation of the Pet Management System can be broken down into several systematic steps. Below are the steps for the system's development from conceptualization to deployment.

1.1.1 Step 1: Requirement Analysis and Planning

- **Identify System Requirements:** Gather functional and non-functional requirements from stakeholders (e.g., pet owners, veterinarians, administrators).
- **Define Features:** Determine all the features required for the system (e.g., service booking, pet profiles, admin panel, product listings).
- **Technology Stack Selection:** Choose the technology stack for both frontend and backend (HTML, CSS, JavaScript, Bootstrap for frontend; PHP, Node.js, Django/Flask for backend).
- **Database Design:** Plan the database structure (e.g., MySQL, MongoDB) to manage users, pets, appointments, and products.

1.1.2 Step 2: User Interface (UI) Design

- **Wireframe Design:** Create wireframes for each page (Home, About, Services, Products, Blog, Contact, etc.) to visualize the layout and user flow.
- **UI/UX Design:** Develop the actual visual design, focusing on usability and aesthetics, ensuring a user-friendly interface. Use tools like Figma or Adobe XD for designing mockups.
- **Responsive Design:** Make sure the design works across all devices (desktop, tablet, mobile).

1.1.3 Step 3: Frontend Development

- **HTML/CSS/JavaScript Development:** Develop the structure and style of the website using HTML and CSS. Use JavaScript for dynamic content like the testimonial slider and appointment booking system.

- **Implement Bootstrap:** Use Bootstrap for responsive design to ensure the system works well across different screen sizes.
- **Frontend Interaction:** Integrate interactivity features such as form validation, service selection, and the "Add to Cart" functionality.

1.1.4 Step 4: Backend Development

- **Database Setup:** Set up the database (e.g., MySQL or MongoDB) to store pet details, user profiles, service information, appointment data, and product listings.
- **Server-Side Logic:** Implement server-side functionality using PHP, Node.js). This includes handling user registrations, appointments, and CRUD operations for managing pets, services, and products.
- **API Development:** If needed, create APIs for third-party integrations (e.g., for payment gateways, user authentication).

1.1.5 Step 5: Integration

- **Frontend-Backend Integration:** Connect the frontend UI with the backend services. Ensure that data is correctly fetched from the database (e.g., pet profiles, service details).
- **Payment Gateway Integration:** If applicable, integrate a secure payment gateway for booking appointments or purchasing products.
- **User Authentication:** Implement login and registration functionality for both pet owners and admin users.

1.1.6 Step 6: Testing

- **Unit Testing:** Test individual components such as forms, buttons, and backend APIs to ensure they function as expected.
- **Integration Testing:** Test the complete flow, such as booking an appointment or placing an order, ensuring all components work together seamlessly.
- **User Testing:** Conduct user acceptance testing (UAT) to ensure the system meets the needs of the pet owners and administrators.
- **Bug Fixing:** Resolve any issues found during testing.

1.1.7 Step 7: Deployment

- **Set Up Hosting:** Choose a hosting platform (e.g., Heroku, GitHub Pages, AWS) and deploy the frontend and backend.
- **Domain Configuration:** If applicable, purchase and configure a custom domain for the website.

- **Final Testing on Live Server:** Perform final checks to ensure the system works on the live server.

- **Monitor Performance:** Monitor server performance to ensure smooth operation and scalability.

1.1.8 Step 8: Maintenance and Updates

- **Bug Fixes and Updates:** Continuously improve the system by fixing bugs, adding new features, and improving performance.

- **User Feedback:** Collect feedback from users to enhance the system in future updates.

Chapter 2

Which Model are Suitable or not Suitable for our project

2.1 Introduction

The Pet Management System is a complex software solution that requires flexibility, iterative development, and continuous feedback. After evaluating several system implementation techniques, it is evident that **Agile development** is the most suitable technique for this project. This document justifies why Agile is the best fit for the Pet Management System by comparing it with other methodologies and highlighting its advantages.

2.2 Why Agile Development is Suitable

2.2.1 Iterative and Incremental Approach

Agile is built around the concept of iterative and incremental development, meaning the system is developed in small, manageable pieces, also known as sprints. This allows the system to be developed and delivered progressively. For the Pet Management System, this means:

- Core functionalities like pet profiles, appointment booking, and service management can be developed and delivered first.
- Features can be released in increments, allowing early engagement with users (e.g., pet owners, administrators) while other features are still in development.
- The system evolves over time, enabling continual improvements based on user feedback.

2.2.2 Flexibility to Adapt to Changing Requirements

Agile development allows for quick adaptation to changing requirements, which is essential in projects where user feedback and business needs evolve. In the case of the Pet Management System:

- New features or adjustments can be added or modified as the project progresses, such as adding advanced analytics or integrating new payment gateways.

- Agile supports constant collaboration with stakeholders, ensuring that any emerging needs are immediately addressed.

2.2.3 Early Delivery of Core Features

Agile promotes delivering functional parts of the system early in the development cycle, which benefits the Pet Management System in the following ways:

- Pet owners and administrators can start using the core features such as user registration, pet profiles, and appointment booking as soon as they are ready.
- This early release provides real-world feedback on the usability and functionality of the system.
- Early releases allow developers to identify bugs or issues in core functionalities while the project is still in its early stages.

2.2.4 Continuous Feedback and Improvement

Agile's emphasis on continuous testing, user feedback, and iteration ensures that the system is refined based on actual usage. For the Pet Management System:

- Users (pet owners, administrators) can provide feedback at the end of each sprint, ensuring the system aligns with their needs.
- The feedback loop leads to continuous refinement and improvement of the system, making it more user-centric.
- Regular testing ensures that bugs are identified and fixed early, resulting in a more stable system at deployment.

2.2.5 Risk Management

Agile provides better risk management because the project is divided into smaller, manageable tasks. This allows the development team to:

- Identify and address potential risks early in the process (e.g., security vulnerabilities, performance issues).
- Adjust the scope or timeline if issues arise, reducing the likelihood of project delays or failures.
- Manage workload effectively to prevent overburdening the development team.

2.3 Why Other Implementation Techniques Are Less Suitable

2.3.1 Waterfall Model

The Waterfall model is a linear approach where each phase must be completed before moving to the next. This methodology is less suitable for the Pet Management System due to the following reasons:

- **Delayed User Feedback:** In Waterfall, the entire system must be developed before it can be tested by users. This means users will have to wait until the system is fully developed, delaying feedback and user engagement.
- **Limited Flexibility:** Once a phase is completed in Waterfall, it is difficult to go back and make changes. This limits flexibility, especially when requirements change during the development process.

2.3.2 Prototyping

Prototyping is a technique where a simplified version of the system is developed to gather user feedback. While this approach is beneficial for user interface design, it is not suitable for the full system lifecycle of the Pet Management System because:

- **Limited Functionality:** Prototypes typically only include a subset of features (usually the UI), and may not address all system components such as database management, security, and server-side logic.
- **Ambiguity in Development Path:** Prototyping can lead to ambiguity about how to move forward with the full system, often resulting in rework or confusion as development transitions from prototype to complete system.

2.4 Conclusion

In conclusion, **Agile development** is the most suitable implementation technique for the Pet Management System. It allows for iterative and incremental development, quick adaptation to changing requirements, early delivery of core features, continuous user feedback, and effective risk management. These advantages make Agile the best choice for ensuring the system meets user needs while remaining flexible and scalable. Other methodologies such as Waterfall and Prototyping, while useful in certain scenarios, do not provide the same level of flexibility and user engagement as Agile does.