

# **JAVA PETSTORE: A CASE STUDY**

**By**

**ROHIT NAMBIAR**

**B.E., Bhartidasan University, India, 2002**

---

**A REPORT**

**Submitted in partial fulfillment of the**

**Requirements for the degree**

**MASTER OF SCIENCE**

**Department of Computing and Information Sciences  
College of Engineering**

**KANSAS STATE UNIVERSITY  
Manhattan, Kansas**

**2005**

**Approved by:**

**Major Professor  
Dr. William J. Hankley**

## **ABSTRACT**

The Java PetStore application (an open source application) is Sun Microsystem's model example of a web application. The Java PetStore is an implementation of a distributed application according to the J2EE blueprints maintained by Sun and its source code can be freely downloaded from the Sun Website. The application consists of a Web site through which it presents an interface to customers where the customers can buy pets online. According to "Sun Microsystem's Java Blueprints", the Java Pet Store application demonstrates certain models and design patterns and documents best practices and architectural ideas for J2EE applications. The application also helps developers and architects understand J2EE technologies, and how each of the J2EE platform components fit together. But the main problem in Sun's primary J2EE blueprint application lies in the poor documentation that fails to explain in detail how the design goals, architectural design/patterns are mapped on to a real web application. Moreover giving just the source code for the application devoid of any class diagrams and lack of enough diagrammatic representation makes it extremely difficult for an end user to understand the design concepts of the PetStore.

The purpose of this case study is to take Sun's primary J2EE blueprint application, the Sun Java Pet Store (Open Source), and study the java petstore applications, its architecture, its respective designs by the way of dividing it into functional modules, representing them in detail with the help of class diagrams and analyzing how it presents a good material for a case study. Based on Sun's J2EE best-practice sample application, we learn the recommended design patterns for building flexible, scalable, cross-platform Web-based applications. We have taken into account what quality goal has been achieved by the application and how the design/architecture affected the design goals and vice versa. We also added thoughts about the performance of an equivalent, revised, and fully optimized J2EE Petstore Web application by looking into some testing statistics performed on the application.

# TABLE OF CONTENTS

|  |            |
|--|------------|
| <b>LIST OF FIGURES .....</b>                           | <b>iii</b> |
| <b>LIST OF TABLES.....</b>                             | <b>iv</b>  |
| <b>ACKNOWLEDGEMENT .....</b>                           | <b>v</b>   |
| <b>1 INTRODUCTION .....</b>                            | <b>1</b>   |
| <b>2 BACKGROUND .....</b>                              | <b>2</b>   |
| 2.1 Webservice Applications .....                      | 2          |
| 2.1.1 Architecture of J2ee Application.....            | 2          |
| 2.2 J2ee Design Patterns Blueprints.....               | 4          |
| <b>3 PETSTORE ARCHITECTURE AND DESIGN.....</b>         | <b>4</b>   |
| 3.1 General PetStore Overview.....                     | 4          |
| 3.2 Requirement & Specifications .....                 | 7          |
| 3.3 Model View Controller Architecture.....            | 8          |
| 3.4 Structural View Architecture.....                  | 10         |
| 3.4.1 Modular Design and its Implementations.....      | 12         |
| 3.4.2 Entity Bean and Session Bean Class Diagrams..... | 19         |
| <b>4 PETSTORE IMPLEMENTATION .....</b>                 | <b>21</b>  |
| 4.1 Page Flow Diagram.....                             | 21         |
| 4.1.1 Page Screenshots.....                            | 22         |
| 4.2 Petstore JSP pages.....                            | 25         |
| 4.3 Petstore XML Files.....                            | 26         |
| <b>5 TESTING.....</b>                                  | <b>26</b>  |
| 5.1 Unit Testing.....                                  | 26         |
| 5.2 Java Profiling.....                                | 27         |
| 5.2.1 CPU Profiling.....                               | 27         |
| 5.2.3 Memory Profiling.....                            | 27         |
| 5.3 Load Testing.....                                  | 27         |
| <b>6 EVALUATIONS AND ENHANCEMENTS .....</b>            | <b>28</b>  |
| 6.1 Testing Results.....                               | 28         |