D2HawkeyeServices

Design Specification

NDS

Neurology Diagnosis System

Version 1.0

Revision History

Date	Revision	Description	Name	Approved By
11/29/2008	1.0	Initial version	NDS Team	Bikram L. Shrestha
12/9/2008	2.0	Detailed design included	NDS Team	

Table of Contents

1 Introduction

1.1 Purpose

This document describes the high level and detail design solutions for the product.

1.2 Scope

This document covers the high-level and detail design aspects of the product. These include:

- System architecture
- Static and dynamic structures of the system
- Deployment diagram
- Detail class diagram
- Table schemas

2 High Level Design

The system basically comprises to two major components: Case-based component and the Rule-based component. These two components operate separately to give the expert system solution.

2.1 System architecture

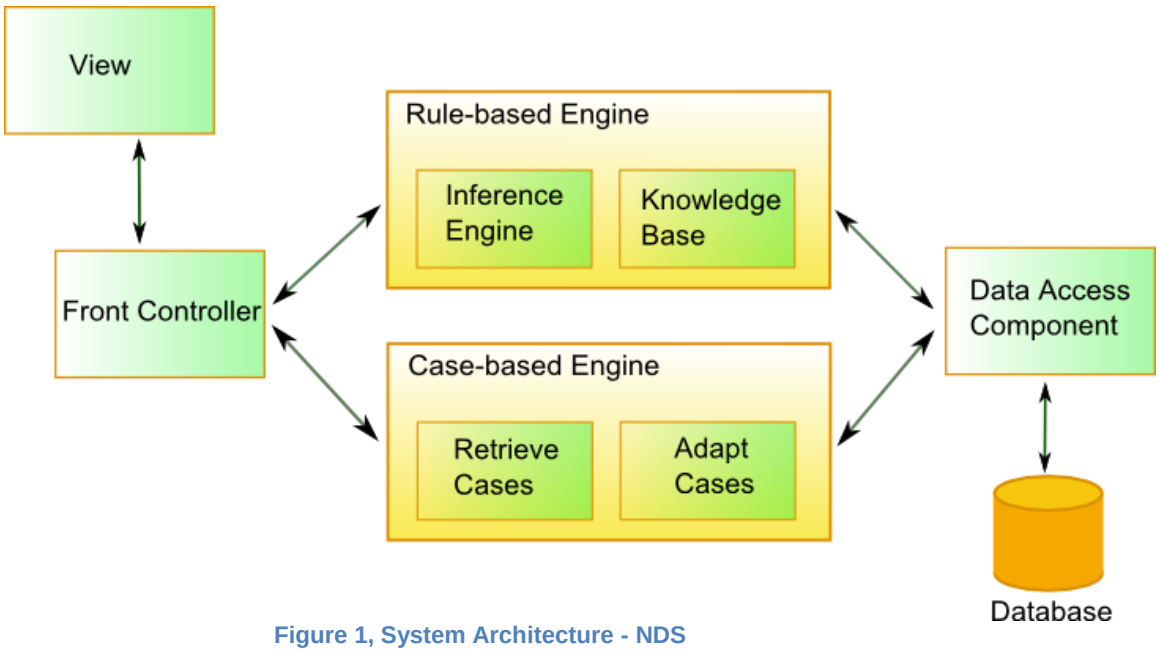


Figure 1, System Architecture - NDS

2.2 Package diagram

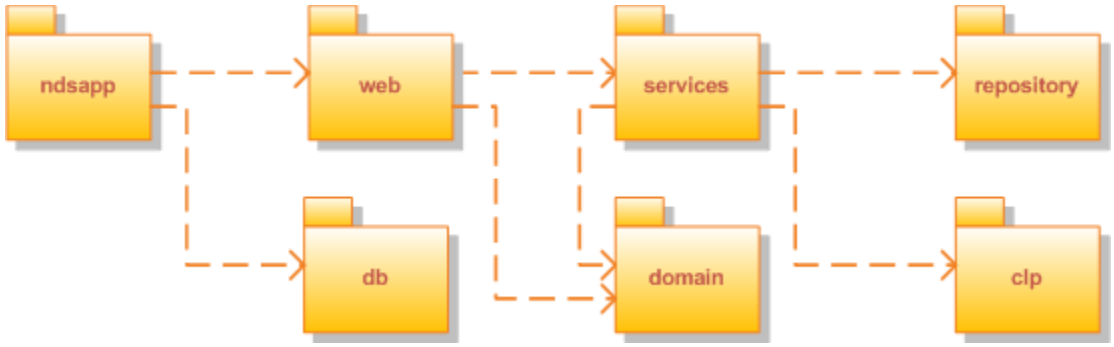


Figure 2, Package diagram - NDS

2.3 Object diagram

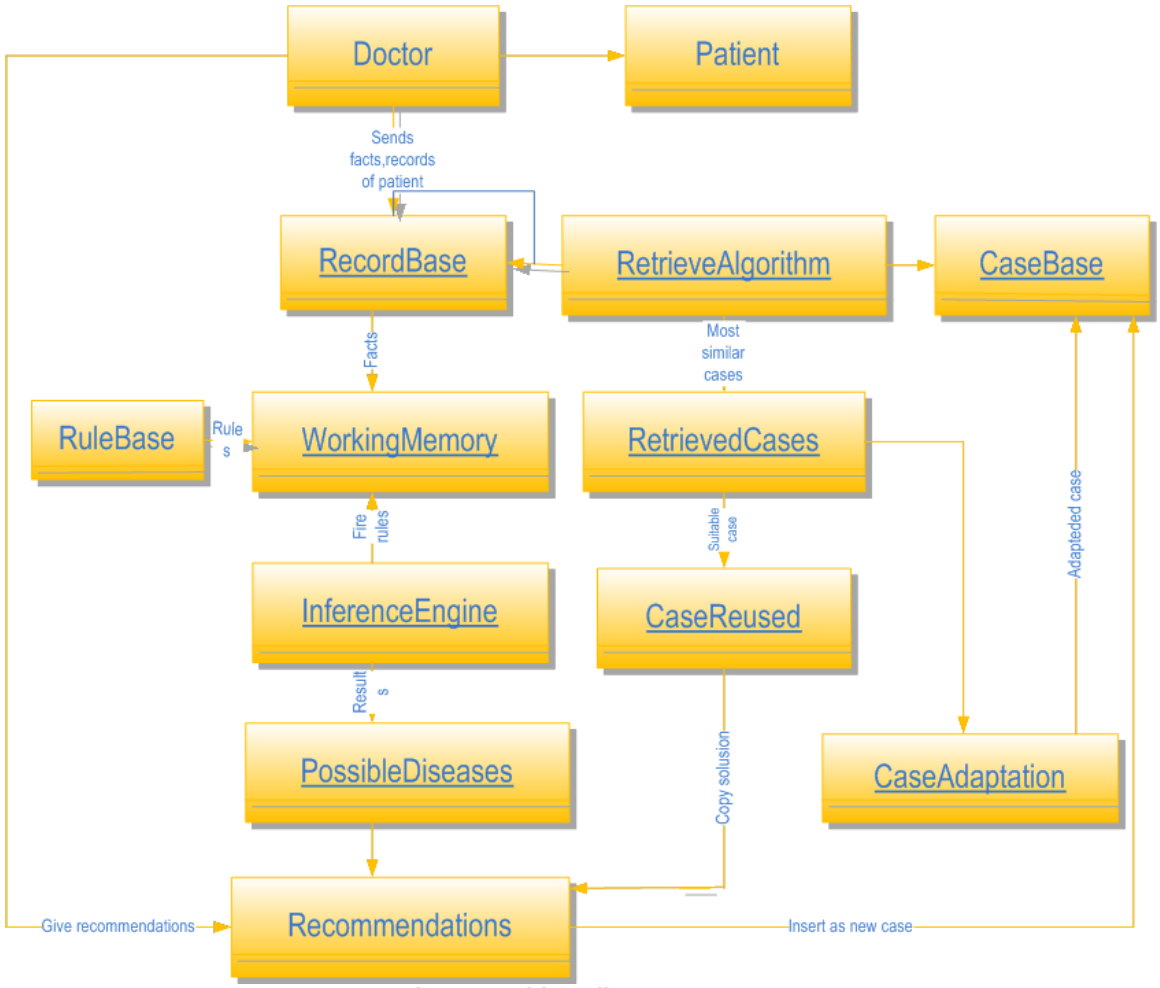


Figure 3, Object diagram - NDS

2.4 ER diagram (list all table names and relationships but do not go into table detail)

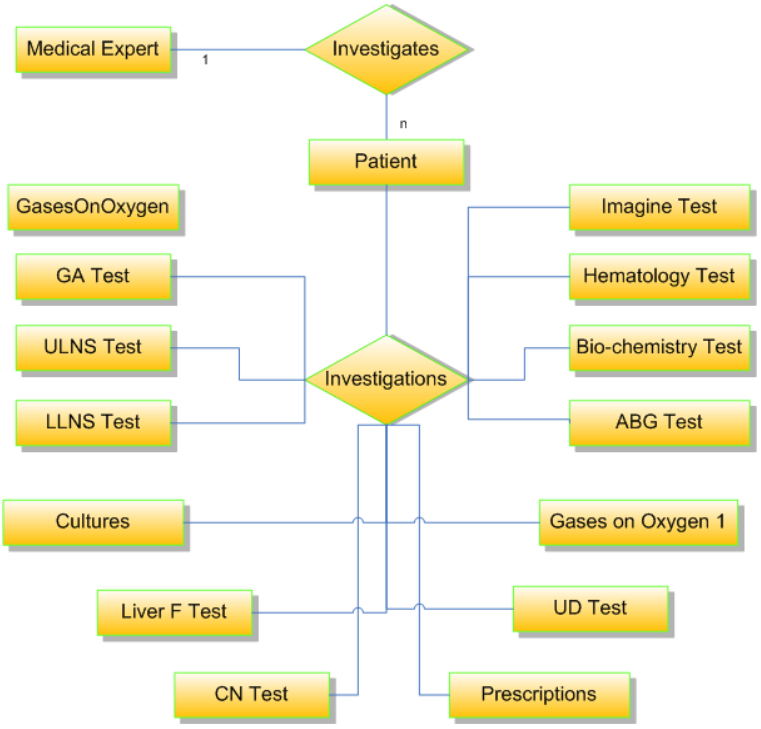


Figure 4, ER Diagram - NDS

2.5 Deployment diagram

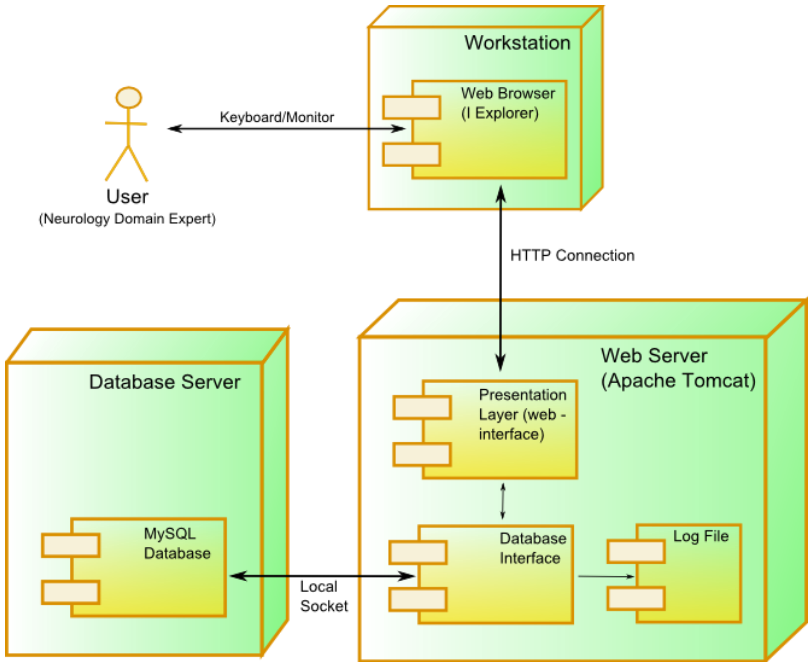


Figure 5, Deployment diagram - NDS

2.6 Frameworks, platforms, and third-party components used

1. Frameworks used
    a. Spring web MVC framework as the web application framework.
    b. JSF (Java Server Faces) to handle the User Interface.

2. Third-party components used
    a. JESS (Java Expert System Shell), as the Inference Engine at http://www.jessrules.com

3. Platforms
    a. Platform independent because the programming language chosen is Java.

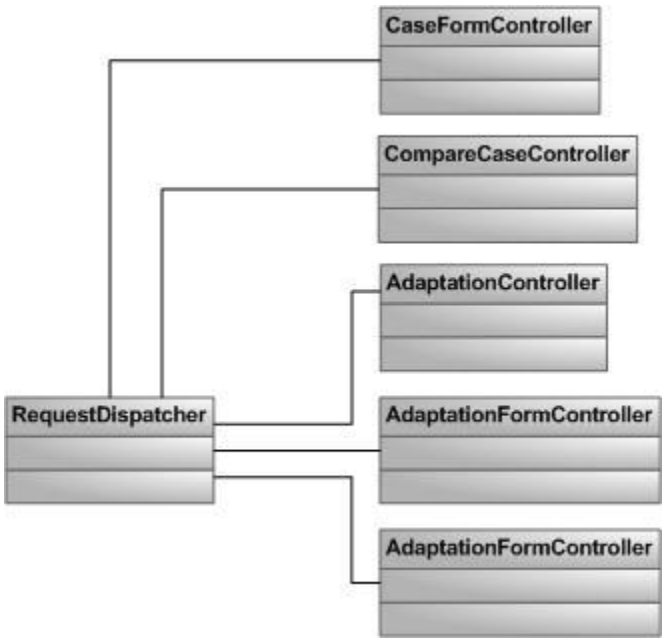
3 Detail Design Specifications

3.1 View

- Purpose
  - Provide a Graphical User Inteface for interacting with the system user.
- Class diagram
  - Not Applicable
- Table structure
  - Not Applicable
- Algorithms used
  - Not Applicable
- Names of source code files that will be produced
  1. index.jsp
  2. searchcases.jsp
  3. casesfound.jsp
  4. caseupdateform.jsp
  5. casedeleted.jsp
  6. caseupdated.jsp
  7. listrules.jsp
  8. ruleupdateform.jsp
  9. ruledeleted.jsp
  10. ruleupdated.jsp
  11. menu.jsp
  12. diseasesinfo.jsp
  13. diagnosis.jsp
  14. expertsystemsolution.jsp
  15. adapt.jsp
  16. caseupdated.jsp
  17. aboutus.jsp
  18. help.jsp

3.2 Front Controller and Page Controllers

- Purpose
  - Receive all the requests from the jsp pages and call the appropriate business logic. Finally, pass the control the appropriate view.
- Class diagram



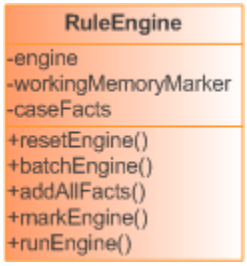
- Table structure
  - Not Applicable
- Algorithms used
  - Not Applicable
- Names of source code files that will be produced
  1. MaintainCasesController.java
  2. SearchCasesController.java
  3. MaintainRulesController.java

3.3 Inference Engine

- Purpose
  - Receive facts and rules to produce expert system solution using the rule-based logic.
- Class diagram
  - Not Applicable
- Table structure
  - Not Applicable
- Algorithms used
  - Not Applicable
- Names of source code files that will be produced
  1. Rule.java
  2. Fact.java
  3. nds.clp
  4. Jess.jar
  5. RuleEngine.java

3.4 Knowledge Base

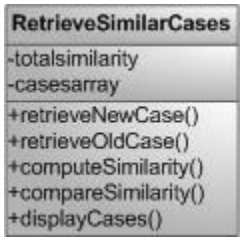
- Purpose
  - Holds the rules in the database as knowledge.
- Class diagram



- Table structure
  - Not Applicable
- Algorithms used
  - Not Applicable
- Names of source code files that will be produced
  - Not Applicable

3.5 Retrieve Component

- Purpose
  - Retrieve similar cases from the database.
- Class diagram



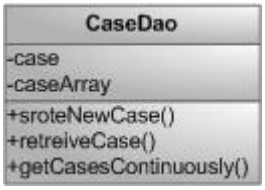
- Table structure
  - Not Applicable
- Algorithms used
  - Nearest neighbor algorithm.
  - Details in the document Nearest *Neighbor Algorithm for Case Retrieval.doc*
- Names of source code files that will be produced
  - Retreive.java

3.6 Adapt Component

- Purpose
  - Provide provision for adaption in reference to the Case-based reasoning cycle.
- Class diagram
  - Not Applicable
- Table structure
  - Not Applicable
- Algorithms used
  - Manually, through the domain expert.
- Names of source code files that will be produced
  1. MaintainCasesController.java
  2. UpdateCase.java
  3. FetchCasesForAdaptation.java
  4. CaseDao.java
  5. Case.java

3.7 Data Access Component

- Purpose
  - Provide methods for database access.
- Class diagram



- Table structure
  - Not Applicable
- Algorithms used
  - Not Applicable
- Names of source code files that will be produced
  1. CaseDao.java
  2. RuleDao.java

3.8 Database Schema Diagram

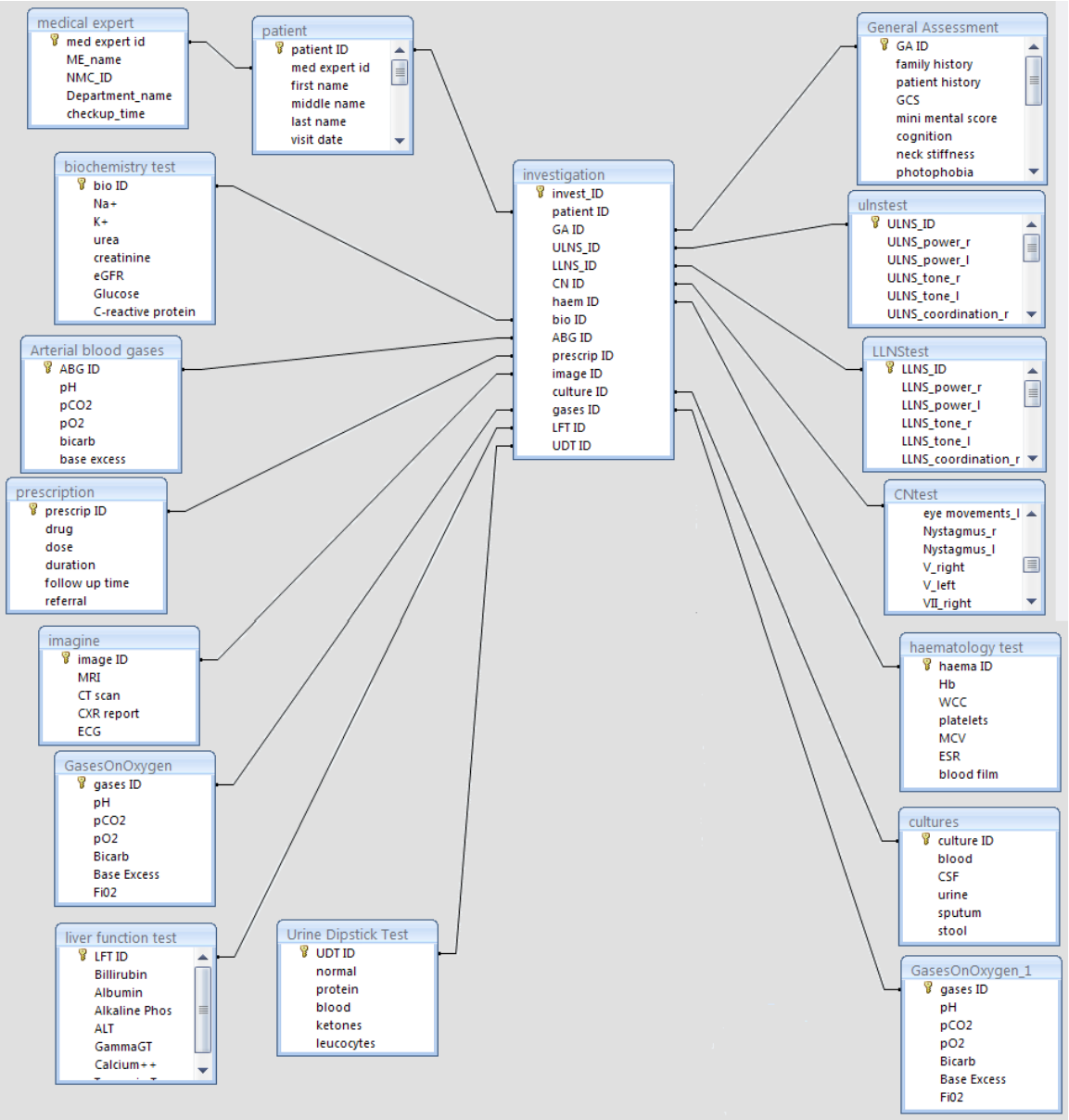


Figure 6 Schema Diagram for the System

3.9 Sequence Diagrams

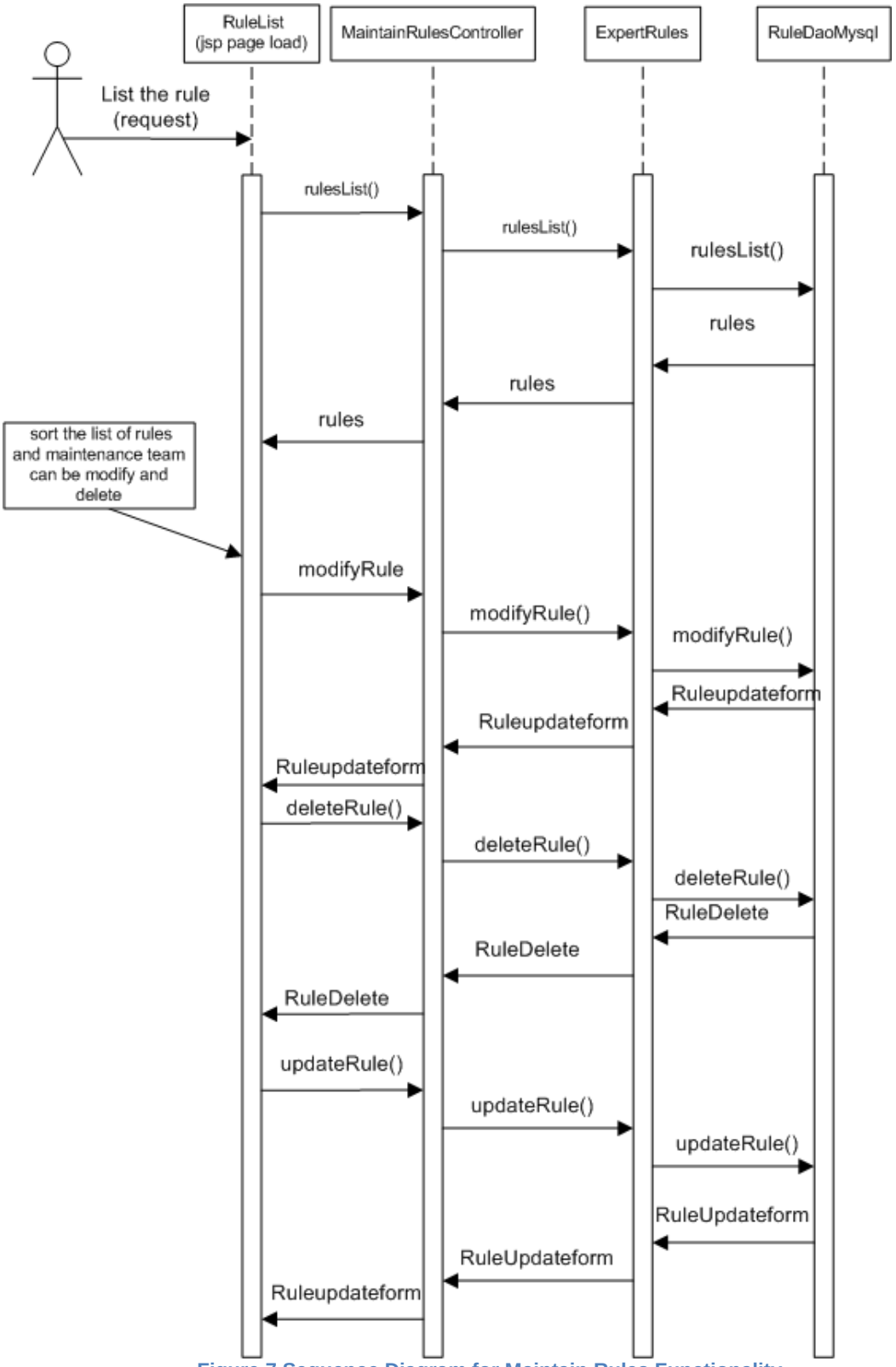


Figure 7 Sequence Diagram for Maintain Rules Functionality

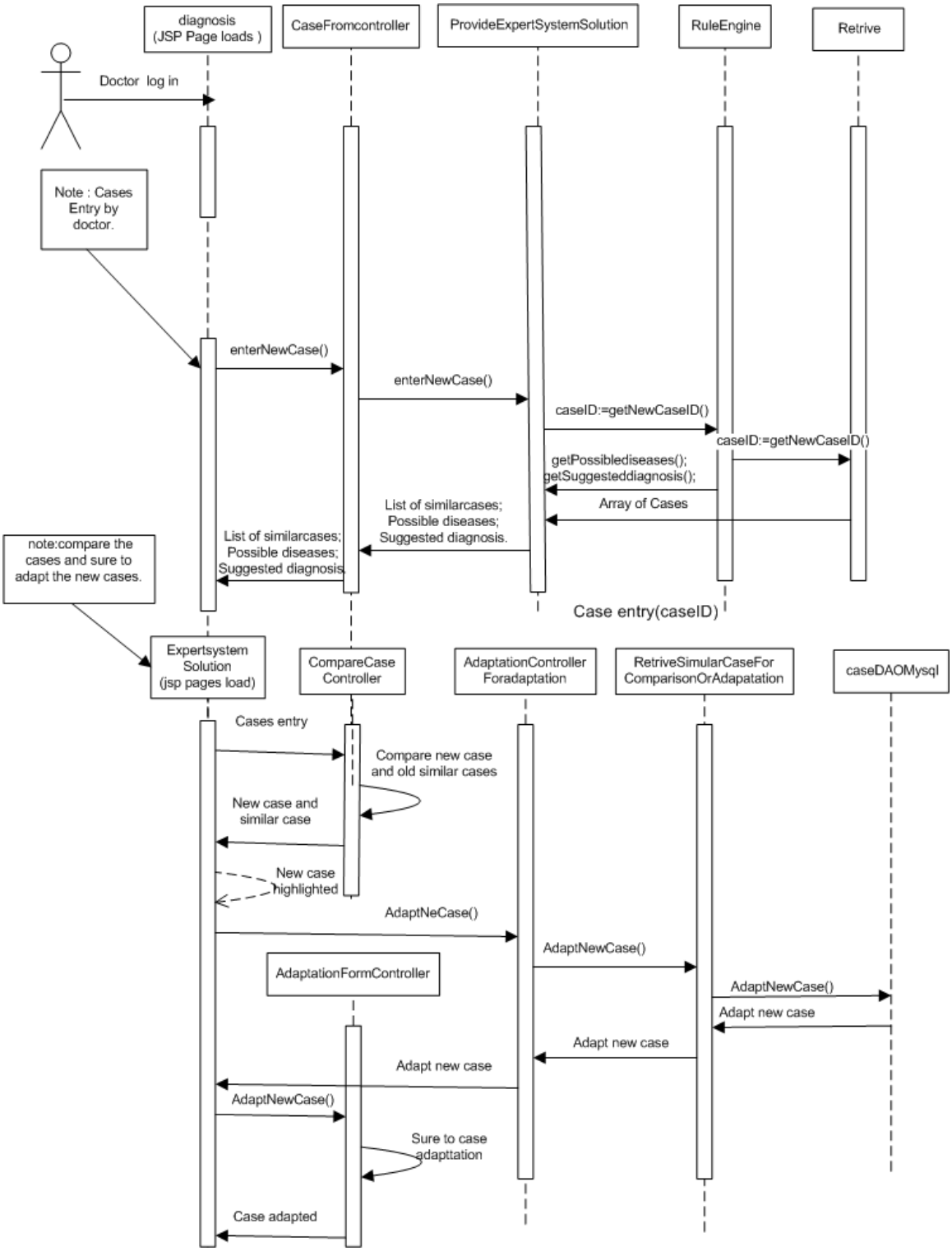


Figure 8 Sequence Diagram for Provide Expert System Functionality

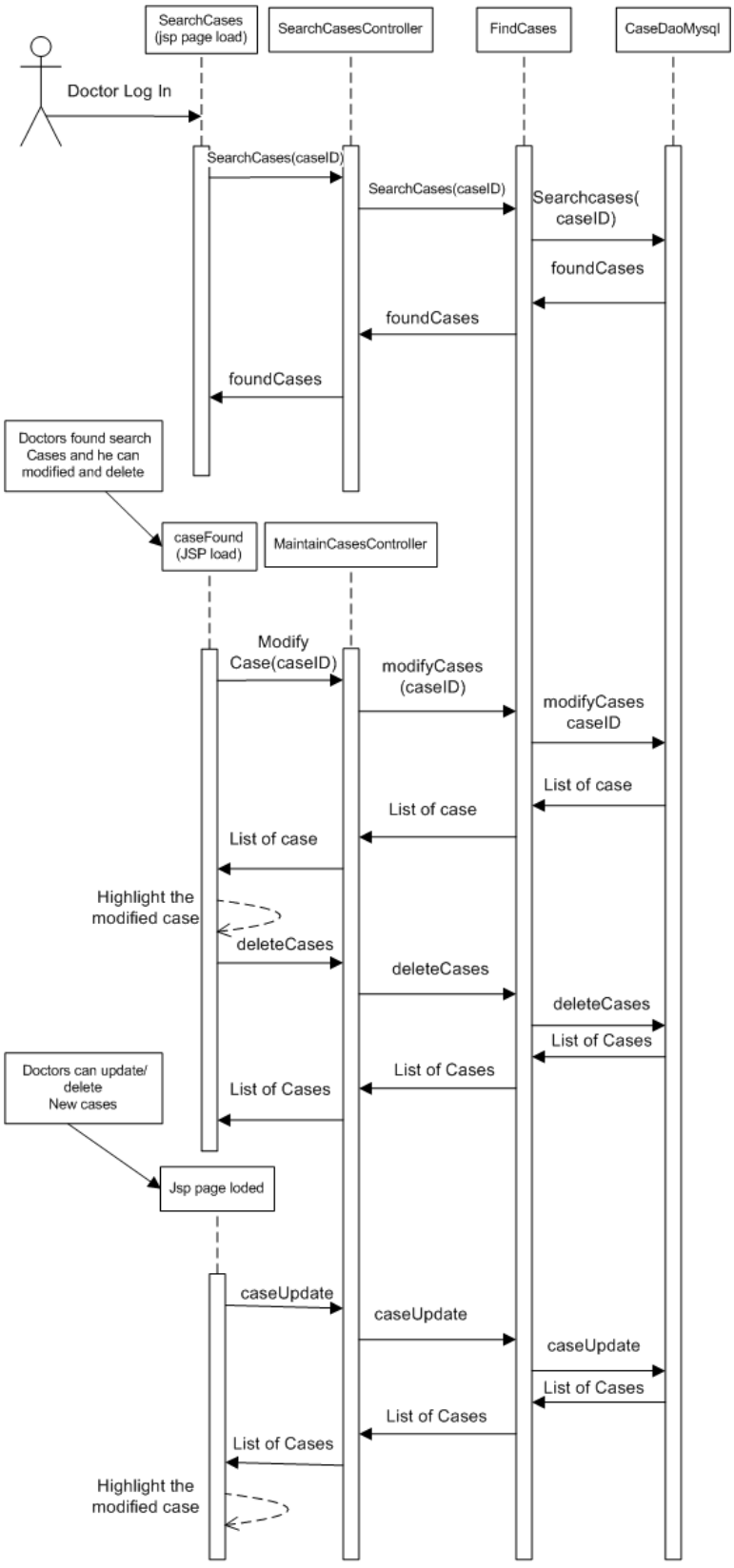


Figure 9 Sequence Diagram for Maintain Cases Functionality

3.10 Detailed Design Showing the Control Flow

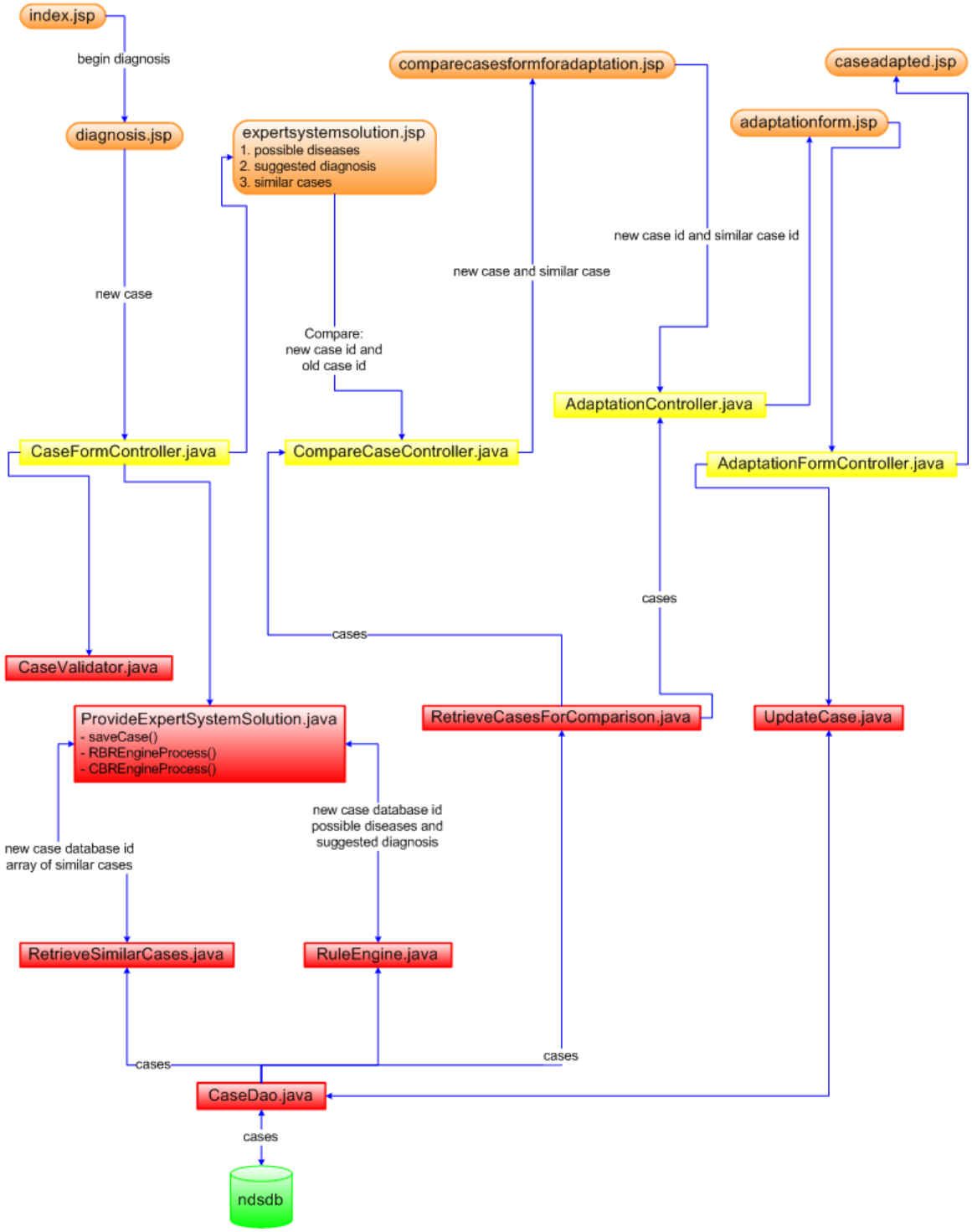


Figure 10 Provide Expert Solution - Functional Requirement

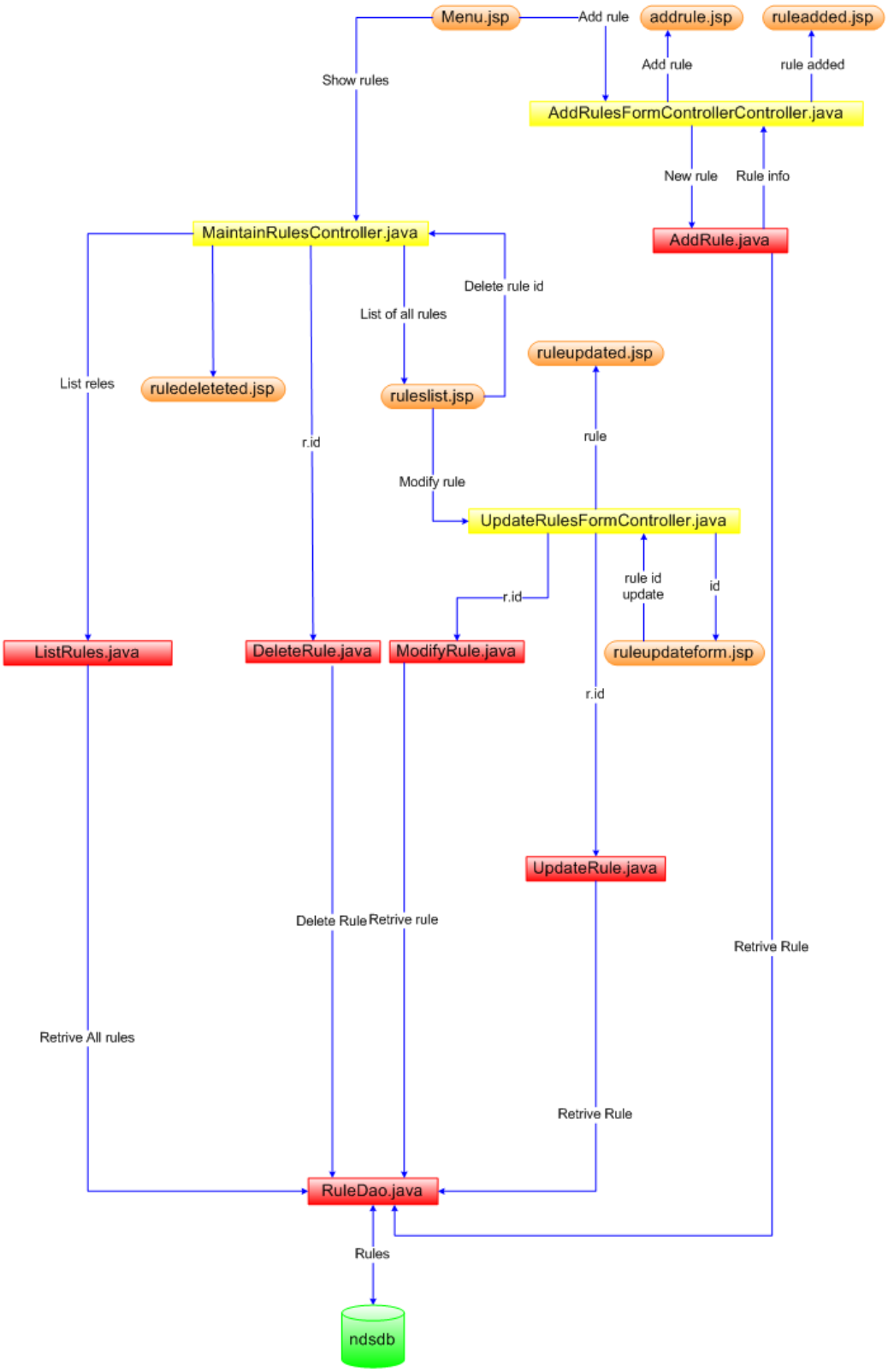


Figure 11 Detailed Design for Maintain Rules Functionality