## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Description</th>
<th>Name</th>
<th>Approved By</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/29/2008</td>
<td>1.0</td>
<td>Initial version</td>
<td>NDS Team</td>
<td>Bikram L. Shrestha</td>
</tr>
<tr>
<td>12/9/2008</td>
<td>2.0</td>
<td>Detailed design included</td>
<td>NDS Team</td>
<td></td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction ........................................................................................................................................... 4
  1.1 Purpose ........................................................................................................................................... 4
  1.2 Scope ............................................................................................................................................... 4
2 High Level Design ................................................................................................................................. 5
  2.1 System architecture ......................................................................................................................... 5
  2.2 Package diagram ............................................................................................................................ 5
  2.3 Object diagram ............................................................................................................................... 6
  2.4 ER diagram (list all table names and relationships but do not go into table detail) ............. 7
  2.5 Deployment diagram ....................................................................................................................... 7
  2.6 Frameworks, platforms, and third-party components used ......................................................... 8
3 Detail Design Specifications .................................................................................................................. 9
  3.1 View ................................................................................................................................................ 9
  3.2 Front Controller and Page Controllers ......................................................................................... 10
  3.3 Inference Engine ............................................................................................................................ 11
  3.4 Knowledge Base ............................................................................................................................. 11
  3.5 Retrieve Component ......................................................................................................................... 12
  3.6 Adapt Component .............................................................................................................................. 12
  3.7 Data Access Component .................................................................................................................. 13
  3.8 Database Schema Diagram ............................................................................................................. 14
  3.9 Sequence Diagrams ......................................................................................................................... 15
  3.10 Detailed Design Showing the Control Flow ................................................................................ 18
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 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

![System Architecture - NDS](image1)

**Figure 1, System Architecture - NDS**

2.2 Package diagram

![Package diagram - NDS](image2)

**Figure 2, Package diagram - NDS**
2.3 Object diagram
2.4 ER diagram (list all table names and relationships but do not go into table detail)

![ER Diagram - NDS](image)

**Figure 4, ER Diagram - NDS**

2.5 Deployment diagram

![Deployment Diagram - NDS](image)

**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 Interface 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. expertsystemsolutions.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 to 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**

```java
public class RuleEngine {
    public void resetEngine()
    public void batchEngine()
    public void addAllFacts()
    public void markEngine()
    public void runEngine()
}
```

- **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**
  ```
  RetrieveSimilarCases
  - totalsimilarity
  - casesarray
  + retrieveNewCase()
  + retrieveOldCase()
  + computeSimilarity()
  + compareSimilarity()
  + displayCases()
  ```
- **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**
  - Retrieve.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>
<thead>
<tr>
<th>CaseDao</th>
</tr>
</thead>
<tbody>
<tr>
<td>- case</td>
</tr>
<tr>
<td>- caseArray</td>
</tr>
<tr>
<td>+ storeNewCase()</td>
</tr>
<tr>
<td>+ retrieveCase()</td>
</tr>
<tr>
<td>+ getCasesContinuously()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RuleDao</th>
</tr>
</thead>
<tbody>
<tr>
<td>- rule</td>
</tr>
<tr>
<td>- ruleArray</td>
</tr>
<tr>
<td>+ storeRule()</td>
</tr>
<tr>
<td>+ getRules()</td>
</tr>
<tr>
<td>+ getRule()</td>
</tr>
</tbody>
</table>
```

- **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