Semantic Web Application
Technical Design Document
Version 1.0

Table of Contents

Conceptual Application Design Document Page 2 of 32

Application Design Document Page 3 of 32

Revision History Version	Date	Description	Author
1.0	1/12/10 12:55 PM	Initial Draft	Ilango Guru
2.0			

1.5 Acronyms, Abbreviations, Terms and Definitions

Please refer to Appendix A for a list of all acronyms and abbreviations.

1 Introduction

1.1 Purpose

The purpose of this document is to outline the technical design of the Semantic Web Application and provide an overview for the Semantic Web Application implementation.
Its main purpose is to -

- Provide the link between the Functional Specification and the detailed Technical Design documents
- Detail the functionality which will be provided by each component or group of components and show how the various components interact in the design
- Provide a basis for the Semantic Web Application's detailed design and development

This document is not intended to address installation and configuration details of the actual implementation. Installation and configuration details are provided in technology guides produced during the course of project.
As is true with any high level design, this document will be updated and refined based on changing requirements.

1.2 Scope

The Application Design outlined in this document builds upon the scope defined in the Requirements phase.

1.3 Document Organization

This document is organized into the following sections: Introduction	Provides information related to this document (e.g. purpose, term definitions etc.)
Design Overview	Describes the approach, architectural goals and constraints, Guiding principles, Java Design patterns used in design and development
Topology Diagram	Describes the various system components and the integration between them
Application Architecture	Describe the application architecture in terms of different layers of application. Description of the presentation layer, business layer, data access layer and resource layer and their relationship to each other.
Object Model	Describes the conceptual representation of the problem domain of an application that embodies the business rules being automated and is usually represented with Class diagram
Database Architecture	Describes the overall Data model for the screening tool
Assumptions and Constraints	Details various assumptions made during design and development of the Semantic Web Application
Appendix A	Describes Acronyms, Abbreviations, Terms and Definitions
Appendix B	Lists all products and tools used in design and development
Appendix C	Lists all the configuration files used in implementation
Appendix D	Describes the data dictionary

1.4 Audience

The intended audiences for this document are XYZ Stakeholders, the project development team, technical architects, database designers, testers and imaginary

2 Design Overview

2.1 Approach

This document is created and extended in multiple phases over the course of the project -

- *Requirements Phase* - During the Requirements Phase, the initial version of this document is created, describing the candidate architecture to be validated in the System Design Phase.
- *System Design Phase* - During the System Design phase, the Evolutionary Prototype is created and this document is finalized by establishing a sound architectural foundation for the Construction Phase.
- *Construction Phase* – During the Construction Phase, this document is not expected to change radically; it is mainly updated to reflect changes in any interface definitions.
- *Transition / Training Phase* – During the Transition/Training Phase, no further additions or modifications are made to this document.

2.2 Architectural Goals and Constraints

The overall architecture goals of the system is to provide a highly available and scalable Web Application for users of the State of Missouri and the State of Delaware, for Pets, Pet Owners, and Pet Lovers.
The Web App can be used in the following ways -

1) <<Refer to the document – Developing the application.pdf>>

A key Architectural goal is to leverage industry best practices for designing and developing a scalable, enterprise-wide J2EE application. To meet this goal, the design of the Web App will be based on core J2EE patterns as well as the industry standard development guidelines for building it.

2.3 Guiding Principles

Guiding principles provide a foundation upon which to develop the target architecture for the screening tool, in part by setting the standards and measures that the application must satisfy. These in turn drive design principles that can be used to validate the design and ensure that it is aligned with Rajesh and Ilango's standards.
Some of the guiding principles that will be followed during the Application's design and development are outlined below.

2.3.1 Scalable

Scalability is the ability of the platform to scale both up and down to support varying numbers of users or transaction volumes. The application should be able to scale horizontally (by adding more servers) or vertically (by increasing hardware capacity or software efficiency).

2.3.2 Flexible

Flexibility is the ability of the application to adapt and evolve to accommodate new requirements without affecting the existing operations. This relies on a modular architecture, which isolates the complexity of integration, presentation, and business logic from each other in order to allow for the easy integration of new technologies and processes within the application.

2.3.3 Standards-Based

Portal/Web services will comply with established industry standards. The standards-compliance will not only apply to application development but also to design, platform/infrastructure and other parts of the Semantic Web application. Examples of standards include HTML, XML, J2EE, and JSP.

2.4 Design Patterns

Design patterns are elements of reusable object oriented software. A design pattern catalog is a repository of design patterns. Use of such patterns makes the design of an application transparent. These patterns have been used successfully by developers in their respective fields, and therefore, the pros and cons of the pattern (as well as implementation issues) are known beforehand. All design patterns are reusable and can be adapted to particular contexts.
Some of the design patterns which will be used in the design and development of the Semantic Web Application are

- Front Controller
- Session Façade
- Business Delegate
- Data Access Object
- Value Object

2.4.1 Front Controller

The Front Controller pattern helps to implement a centralized entry point that controls and manages user (screen) request handling. The controller manages the handling of the request, including invoking security services such as authentication and authorization, delegating business processing, managing the choice of an appropriate view, handling errors, and managing the selection of content creation strategies.

2.4.2 Session Facade

The Session Façade pattern (using a session bean as a façade) helps to encapsulate the complexity of interactions between the business objects participating in a workflow. It manages the business objects, and provides a uniform coarse-grained service access layer to clients, that expose only the required interfaces.

2.4.3 Business Delegate

The Business Delegate pattern helps to reduce coupling between presentation-tier clients and business services. The Business Delegate hides the underlying implementation details of the business service, such as lookup and access details of the EJB architecture. Technical Design Document Page 8 of 32

2.4.4 Data Access Object

The Data Access Object pattern helps to decouple the session EJB layer from the database thus increasing the portability of the application.

2.4.5 Value Object

The Value Object design pattern, also known as the Data Transfer Object, efficiently transfers remote, fine-grained data by sending a coarse-grained view of the data. This design pattern will be used for the communication between the middle tier and the back end.

2.5 Design Principles

Best practices and design principles will be applied in two main areas –
1) Presentation Services to individual desktops should be uncoupled -
   a) Presentation services are delivered to a web browser rather than to custom client software. A range of modern browsers that support HTML, DHTML, and XML are required.
   b) A common look and feel for our Semantic Web application.
   c) Client side JavaScript should be used for validating user input and prevent round trips between the browser and the server
   d) The Pet Information Interface will be designed in such a way that common user interface functionality will be implemented in a similar manner across the board. Examples of this include –
      • A consistent way of capturing date inputs
      • A uniform way of displaying informational and error messages to the users
      • A uniform way of displaying required and optional fields in the screens.
2) Business Rules should be encoded within the application development framework-
   a) Business rules will need to be separated from the presentation and database frameworks
   b) Server applications are based on event-based systems. Complex server side event cascades will need to be supported.
   c) Standard frameworks for encoding business rules and events will need to be used.
   d) Adoption of a component based framework needs to be considered to promote reuse of information objects.

3 Topology Diagram

The diagram below provides a illustration of the System Architecture along with various system components that will be used in architecting the Semantic Web Application -

Fig 1: Topology Diagram

<<Diagram here>>

Interaction of software components along with its responsibilities is explained below -

Web Server – Web server is responsible for serving web pages, mostly HTML pages, via the HTTP protocol to clients. The Web server sends out web pages in response to requests from browsers. A page request is generated when a client clicks a link on a web page in the browser.
JBoss/WebSphere Application Server – Application server hosts the Online Screening application and hosts the business logic and the business model classes of applications. It serves requests for dynamic HTTP web pages from Web servers.
Oracle 10g Screening Database – Screening database stores the screening data, program and question

information, audit trails of screening application in relational format.
HTTP - Hyper Text Transport Protocol is the communication protocol used to connect to servers on the World Wide Web. The primary function of HTTP is to establish a connection with a Web server and transmit HTML pages to the user's browser.

JDBC – Java Database Connectivity is an application program interface (API) specification for connecting programs written in Java to the data in popular databases. The application program interface lets you encode access request statements in structured query language (SQL) that are then passed to the program that manages the database. It returns the results through a similar interface.

XML - A programming language/specification developed by the W3C, for organizing and tagging elements of a document so that the document can be transmitted and interpreted between applications and organizations. Semantic Web Application.

c) Client side JavaScript should be used for validating user input and prevent round trips between the browser and the server

d) The Semantic Web Application user interface will be designed in such a way that common user interface functionality will be implemented in a similar manner across the board. Examples of this include –

- A consistent way of capturing date inputs
- A uniform way of displaying informational and error messages to the users
- A uniform way of displaying required and optional fields in the screens.

2) Business Rules should be encoded within the application development framework-

a) Business rules will need to be separated from the presentation and database frameworks

b) Server applications are based on event-based systems. Complex server side event cascades will need to be supported.

c) Standard frameworks for encoding business rules and events will need to be used.

d) Adoption of a component based framework needs to be considered to promote reuse of information objects.

4 Application Architecture

Application architecture defines the various components and their interactions in context of a whole system. Application architecture is the critical software that bridges the architectural gap between the application server and the application's business logic, thereby eliminating the complexities and excessive costs of constructing, deploying and managing distributed enterprise applications.

The Semantic Web Application will have a layered application architecture which provides some of the key features below –

STRUCTURE: Organizing applications along business-level boundaries and not technical boundaries SPEED & FLEXIBILITY: Making application changes through configuration and not programming CONTROL: Modifying, extending or overwriting any architectural element. REUSE: Achieving greater reusability and integration by loosely coupling application logic to infrastructure.

At a *conceptual* level, they represent distinct and cohesive aggregations of functionality. The Semantic Web Application design is based on a tiered approach. "A tier is a logical partition of the separation of concerns of the system. Each tier is assigned its unique responsibility in the system. We view each tier as logically separated from one another. Each tier is loosely coupled with the adjacent tier." The Semantic Web Application architecture can be represented in the following layers illustrated by the diagram below:

<<Diagram here>>

Fig 2: Application Architecture Overview

Application architecture defines the various components and their interactions in context of a whole system. Application architecture is the critical software that bridges the architectural gap between the application server and the application's business logic, thereby eliminating the complexities and excessive costs of constructing, deploying and managing distributed enterprise applications.

4.1 **Presentation and Content Layer**

The Client Tier represents the point at which data is consumed by the system's users which include online users as well as external systems.

4.1.1 Presentation Layer

A standard Internet Browser such as Internet Explorer is the primary client for the online Screening application. HTML pages are delivered to the client browser by the Screening application upon a user request. The Web Pages also include JavaScript functions where applicable. If JavaScript is turned-off, server-side validations are performed to ensure all validations are met. The Web Pages of the Screening application will conform to the American Disability Act, U.S. Section 508.

4.1.2 Content Layer

The content layer as the name signifies is the front-end information layer that the end-user interacts with. Data-to-content conversion and Content-to-data conversion are the two primary responsibilities of this layer. Any application that is created will use the common framework components to implement the primary responsibilities using the technology that seem most appropriate for that application. Choice of technology for this layer would range from plain HTML to a Java-HTML combination to a smart applet or an application.

4.2 Business Objects Layer

The Business layer will implement the business rules for the application. It will host the business service components as well as business objects (BO). These Business Services include Enterprise Java Beans and the BO's include the dependent JAVA classes that will provide service API's to the business rules and operations required by the application. Business components are software units, and process business logic. The business components will implement the following:

- Business rules, such as calculations and validations
- Interfaces between the user interface and the resource layer

The business logic layer will run under the "Application Server" environment. Application Servers provide support for transaction control, thread management and other run-time services that make application development much simpler and more reliable. Business components are generally computation-intensive. They will use Data Access objects (DAO) to communicate with the database. The Business layer will constitute of:

- Java Beans and Java Classes: Java Beans are used to manage the data flow between the layers. Java classes on the other hand are simple java objects that provide utilities to the application. They may also contain business logic and provide other supporting services
- Enterprise Java Beans (EJB): Enterprise Java Beans is the server-side component architecture of the J2EE standard. The EJB's house the business components and reside on the application server. There are two types of Enterprise Java Beans:
    - Session EJBs: Within Session EJBs there are two sub-types – Stateful Session Beans and Stateless Session Beans. These components are typically used to process business logic and commonly referred to as business service components.
    - **Entity EJBs: are used as a persistence mechanism. Entity beans also consist of two types – Container managed persistence or Bean managed persistence. Entity EJBs are mentioned here only for completeness, they will however not be used in the design of the Semantic Web Application due to performance concerns. Plain-Old-Java-Objects (POJOs) will be used in the form of Data Access Objects (DAOs) to perform persistence and as such is a preferred approach to persistence. <<We should not be using Entity EJBs for persistence. Instead Hibernate/JPA should be used**

4.3 Data Access Layer

Data Access Objects using Java Database Connectivity (JDBC) will manage the interface to the database. Persistence can be complex in large applications using protocols like JDBC. Neither the client nor the business component needs to be aware of this complexity. Moreover there are many forms of storage from databases, to flat files. Decoupling the persistence logic from the business components and client allows for a flexible, easy to maintain application. The Data Access Object (DAO) pattern allows for the abstraction of the persistence from the business component. The Data Access Object manages the connection to the data source to obtain and store data. It encapsulates all access to the data store.

4.4 Resource Layer

The resource layer includes the underlying resources that the application uses to deliver its functionality. This includes using a Database and file system to persist information.

4.5 Common Applications Framework

4.5.1 Design Principles

The Common Application Framework components provide utility classes that are used across the application. The framework components provide the other application components with certain base functionality that is required for the other components to function.

4.5.2 Reference Table Architecture

An important framework module is the Reference Table Architecture. The Reference Table Architecture allows for easy administration of the application by allowing for simple database updates for adding new programs or <<other things>> to the Semantic Web Application. The Reference Tables employ a generic design that allows for various data elements to be cached at application startup. The Reference Table components provide a mechanism to access all information stored within the Reference Tables and make it available to any component within the application. Caching of the data at application startup allows for all the reference data to be accessible in-memory, saving multiple trips to the database. The Reference Table Architecture enables the relatively simple process of adding new programs and questions to the application.

4.5.3 Question Engine

The framework also consists of the Pet Questions Engine which is a key component of the Dynamic Screen Builder discussed above. It is the question engine determines what questions should be displayed to the user depending on the previous input provided by the user. To facilitate the dynamic generation of questions the Question Engine determines the dependency between the questions and accordingly determines what questions are required to be answered by the user. For example, if an pet is a female and is over ‘x’ years of age then the pregnancy question should be displayed to the user. Or the spaying question can be displayed to the user.

4.5.4 Rules Engine

The Rules Engine is utilized for processing declarative business rules and is loaded into memory on Application Startup. The Screening related session beans call the Rules Engine manager to process the declared business rules based on user’s input. Any additional questions that need to be validated as part of the screening application need to be added to the corresponding decision tables which are excel spreadsheets.

4.6 Rules Engine Design

All the business rules related to the Semantic Web application will be defined in MS-Excel spreadsheet.<<Think about alternatives as well>> A conversion utility converts the Decision Tables to XML.

Upon initialization of the system, the Decision Tables and the Entity Description Dictionaries are loaded into Java Structures. The Decision Tables are converted into binary trees, with condition nodes defining the branches in the trees, and Action Nodes defining the leaves. The Entity Description Dictionaries define each possible Entity by type (Individual, Case, etc.) and their possible attributes (age, breed, etc.).
In order to run the Rules Engine for the data in a particular case, an EntityFactory is acquired which allows the data loading program to populate the Entities for that case. The XML for the Entity Description Dictionary defines these Entities, which roughly equate to the tables in the database of cases. The EntityFactory allows the data loader to create instances of the entities defined in the XML (like the case, each individual, the relationship objects that define the relationships between individuals, etc.). Once all of the Entities have been created, and their attributes defined, the main DecisionTable is executed.
Once that DecisionTable returns, the rules engine has completed its evaluations and changes. The data writer then queries the appropriate attributes on the entities, and writes these values back to the database.

For more information regarding Rules Engine, please refer to the Rules Engine Use Case (Determine Potential Eligibility UC-SC-06).

<<Diagram here>>

Fig 3: Rules Engine Design

Underlying technology for Rules Engine application are Visual basic <<think of an alternative>>, Java, XML parsers like Xerces and Xalan, XML and XSLT.

Rules cannot be changed when the application is running. If Rules have to be changed, it is necessary to change Excel spreadsheet and reload the XML file to Application server's memory again.

4.7<<Pet Application>> Sequence of Events

In this section we detail the significant interactions between the major components for the Semantic Web module from a user's click to executing business rules using the rules engine. Below is a high level logical sequence diagram depicting the significant interactions within the Semantic Web application.

Note: The Sequence Diagram below is only a logical representation of the significant interactions of the system and may not directly map to the physical interactions of the system.

4.7.1 User

The user is any authenticated and authorized user of the Online Screening Tool application. The user could be a public user or an employee of an agency.

4.7.2 JSP

The Java Server Pages (JSPs) are the dynamically generated web pages that a logged-in user interacts with.

On the click of a button on the web page, a user triggers a series of actions that are executed.

4.7.3 Controller Servlet

The controller servlet is primarily responsible for receiving an HTTP request when the user clicks a button on the web page. The servlet's service method responds to a user's click in the form of an HTTP request.

4.7.4 Service Controller

Since the user is completing a Screening application and does not require the user's state to be maintained, the controller invokes the appropriate Business Service Session Bean. The Service Controller is also responsible for determining which page the user is forwarded to next. Since there is no dynamic scheduling of pages required based on the user's input the Driver is not invoked. Instead the service controller directly invokes the appropriate stateless session bean associated with the screening module.

4.7.5 Session Bean

The Business Service Stateless Session Beans act as a session façade for the screening module and call the underlying rules engine to process the business rules based on the user's input.

4.7.6 Screening Manager

The screening manager invokes the Rules Engine to process the declarative business rules.

4.7.7 Rules Engine

The Rules Engine is used for processing declarative business rules. It is loaded on application startup and is invoked in-memory by the screening manager. The Rules Engine's *executeDT()* method invokes the decision tables that are in-memory and performs the validations specified in the tables.

4.8 Package Structure View

Package structure depicts the various packages used in Screening application and relationship among them.

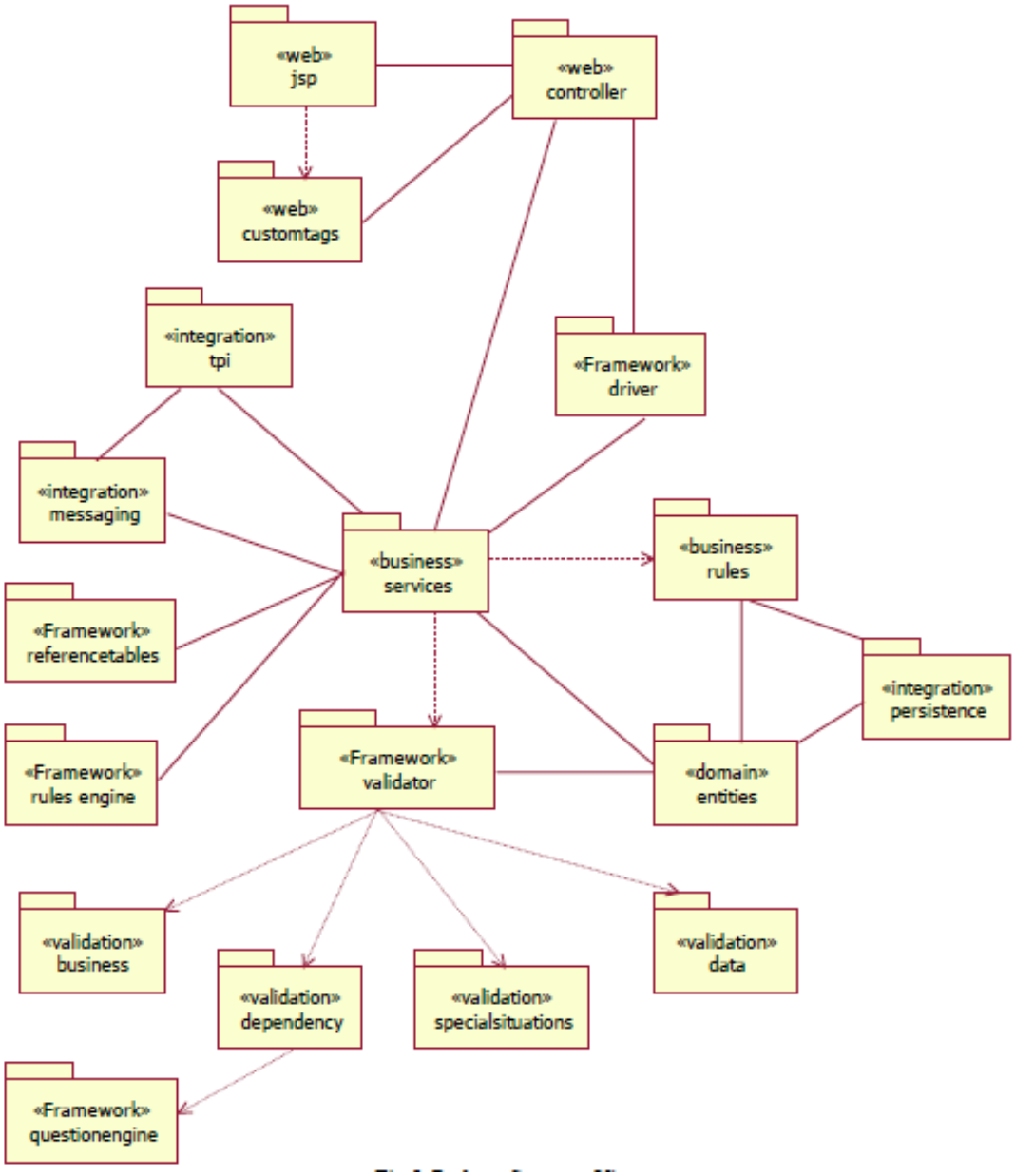


Fig 5: Package Structure View

JSP: This package includes all the Java Server Pages that have developed for the Semantic application which includes both Screening and Intake. The JSPs form the View of the Screening tool and constitutes the primary user interface of the application.

Custom Tags: Custom tags define declarative, modular functionality that can be reused by any JSP page. Tag libraries reduce the necessity to embed large amounts of Java code in JSP pages by moving the functionality of the tags into tag implementation classes. In the Screening application we have designed and developed the question tag which given the question attributes displays a question or a set of questions on a JSP.

Controller: The controller package contains the Controller servlet among other components whose

primary purpose is to provision a dialog with the user. The components within this package determine what pages to display next based on user interaction. This package is used by both the Screening and Intake components of the application.

Reference Table: The Reference Table package contains the components that provide the Table-driven features of the application. For example, if new programs need to be added to screening or intake, simple additions to the database tables is all that is required. On application startup the reference tables are cached in memory and utilized by various application components to perform business services.

Rules Engine: The Rules Engine package contains all components that provide the Rules Engine functionality. It contains components to transform the decision tables to validate business rules used primarily for the screening component of the application.

Question Engine: The Question Engine package contains the components that facilitate the dependency validations. The question Engine also determines which questions appear on the user's screen.

Business Services: The Services package includes the actual validation service which acts as a façade that calls the underlying validation framework to perform the requisite business validations.

Business Rules: The rules package contains the business objects, simple java objects that encapsulate the calls to the data access objects and perform any additional business rules.

Validation: The validation package includes the validation framework which is utilized for business validations. It includes the aforementioned business, dependency, special situations and data validators.

4.9 Object Model

Object Model is the description of the structure of the objects in a system including their identity, relationships to other objects, attributes, and operations.

Object Model for Screening application consist of one or more classes, events, functions, interfaces, methods, namespaces, objects, and properties. Object Model for Rules Engine and Question Engine is depicted below. Detailed Object Model will be submitted at the end of Construction Phase.

4.9.1 Question Engine

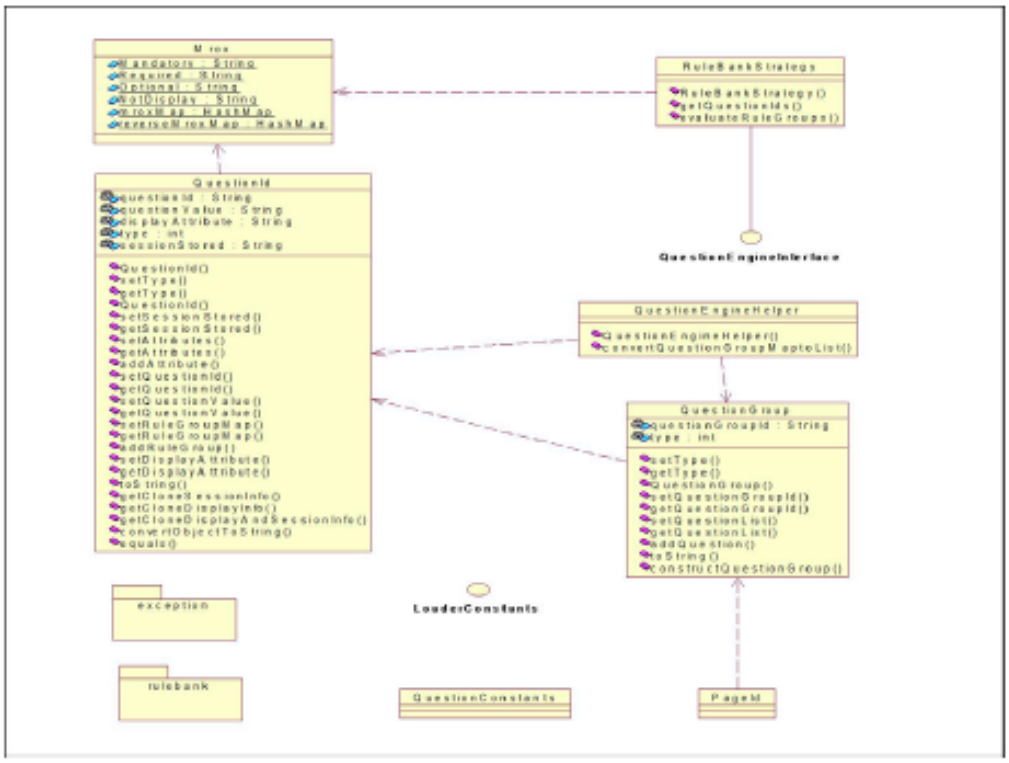


Fig 6: Question Engine

4.9.2 Rules Engine

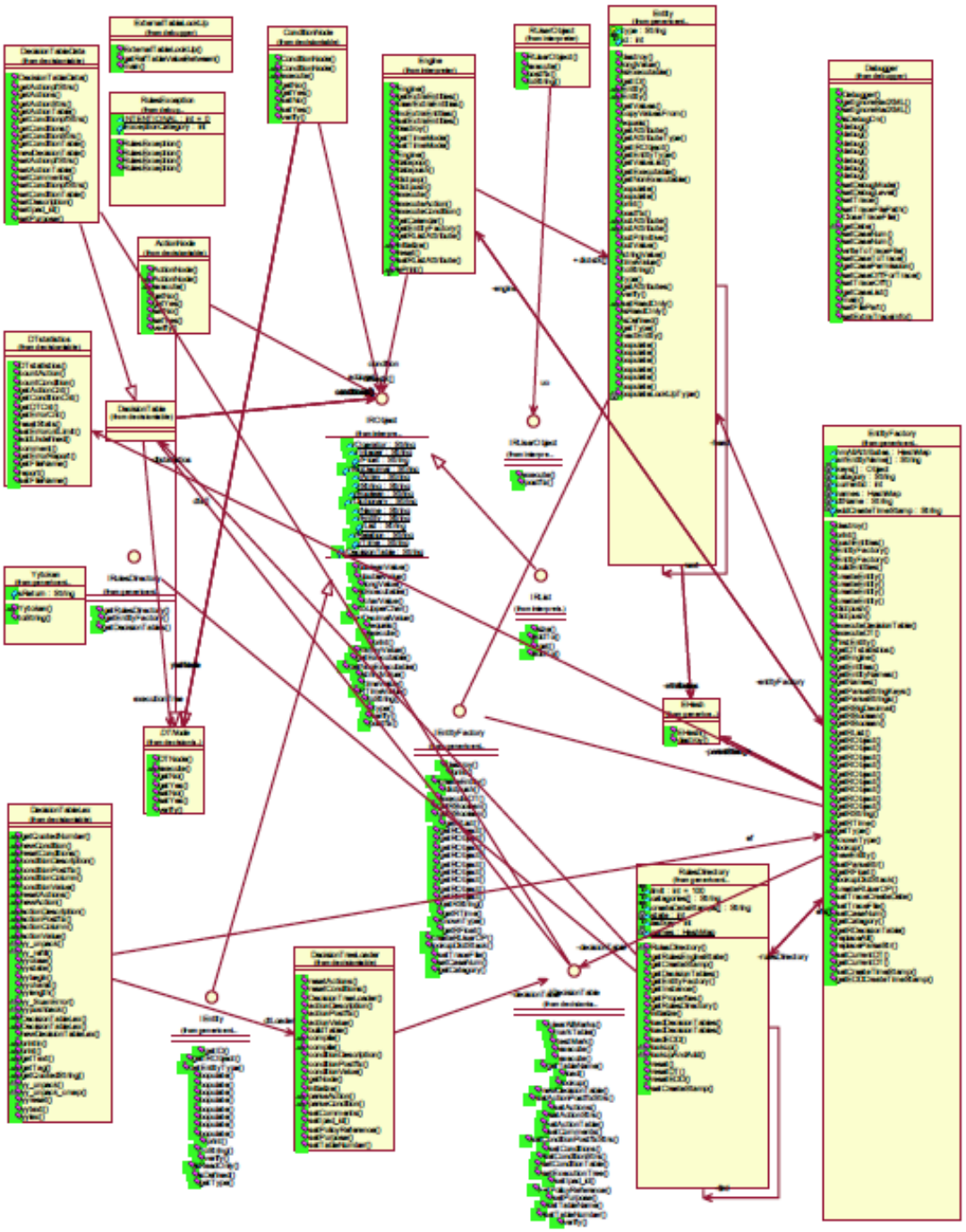


Fig 7: Rules Engine

5 Application Implementation

The Application Deployment Structure is shown below. Screening application will be deployed as EAR (Enterprise Archive) file using ANT build scripts. **Directory Structure:**	
EAR	J2EE applications are packaged as an Enterprise Archive file with .ear extension. EAR file is jar file with .ear extension. Web components and EJB components part of an application will be packaged within an ear file.
JAR	Java Archive is used to package java libraries and EJB components
WAR	J2EE Web applications are packaged as WAR (Web ARchive) with .war extension.

Diagrammatic layout of Semantic Web Application deployment structure is shown below –

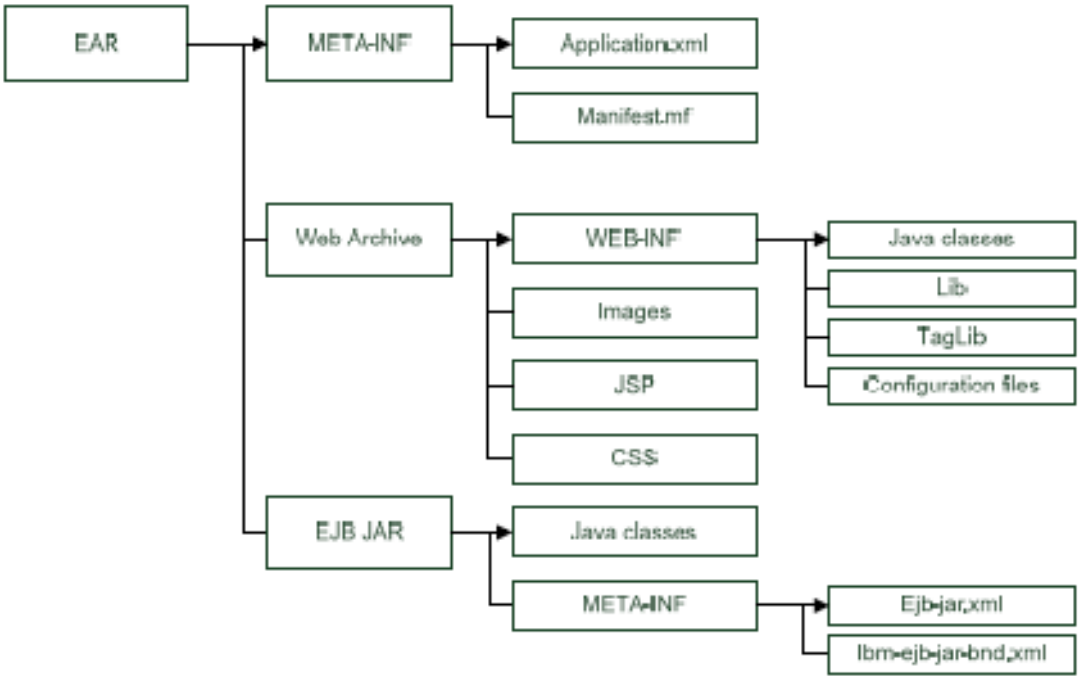


Fig 8: Screening Directory Structure for Deployment

6 Database Architecture

The Sematnic web application will use Oracle 10g Database as its repository. Later ontology and Knowledgebase will also serve as <<repositories>> of information.

Information and data that need to be stored in our Relational database will be determined based on discussions between Stakeholders <<ilango and Rajesh>> and IT staff <<ilango and rajesh>> and also

during Design and Development phases of the project.

6.1 Data Model

Data Model is a method for describing data structures and a set of operations used to manipulate and validate that data. Data Model for the Online Screening application is as shown below –

<<Diagram Here>>

Fig 9: Data Model for Screening tool

6.2 Tables

Screening Database Schema will broadly have three categories of tables -

1. Screening Tables – Tables used for screening, storing questions, possible answers and metadata information - QUESTION, QUEST_CONSTRUCT, QUEST_VALIDATOR, HELP_KEYWORDS, SCR_QUES_REQ, APPLICATIONPROGRAMS, SCR_GENERAL, SCR_HOUSHOLD, SCR_INDIVIDUAL, SCR_RELATIONSHIPS, SCR_FINANCIAL, and SCR_RESULTS.
2. Admin Tables– Tables used for the managing reference data - RT_TABLE, RT_FIELD_DOMAIN, RT_FIELD, RT_FIELD_VALUES
3. Framework tables – Tables used for Page display – FW_PAGE, FW_PAGE_ACTIONS

Detailed Schema design along with data model and field definition will be determined during the ongoing phases. All the table definitions will be documented in the Data Dictionary document (found in *Appendix D* of this document).

6.3 Reporting Solution

The Semantic Web Application Tool Reports will be generated off the Relational Database using SQL queries as illustrated in the diagram below. Some of the sample reports that will be generated out of the application are number of users using the Dog screening application, screening application pass /fail ratio, etc. The data in the SCR_GENERAL and SCR_RESULTS tables will be stored for each screening application.

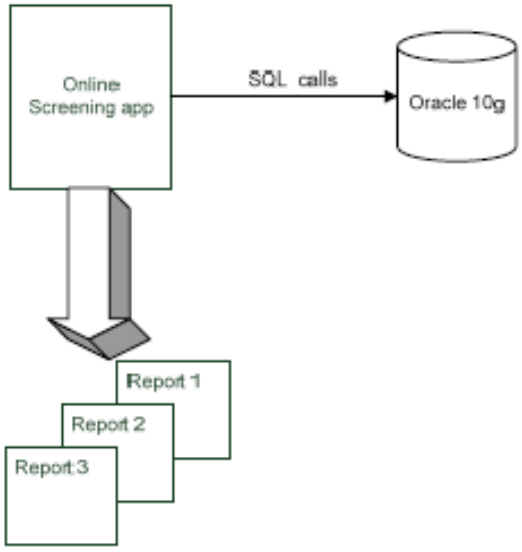
6.2 Tables

Screening Database Schema will broadly have three categories of tables -

1. Dog Information Tables – Tables used for Dog screening, storing questions, possible answers and metadata information - QUESTION, QUEST_CONSTRUCT, QUEST_VALIDATOR, HELP_KEYWORDS, SCR_QUES_REQ, APPLICATIONPROGRAMS, SCR_GENERAL, SCR_HOUSHOLD, SCR_INDIVIDUAL, SCR_RELATIONSHIPS, and SCR_RESULTS.
2. Admin Tables– Tables used for the managing reference data - RT_TABLE, RT_FIELD_DOMAIN, RT_FIELD, RT_FIELD_VALUES

3. Framework tables – Tables used for Page display – FW_PAGE, FW_PAGE_ACTIONS

Detailed Schema design along with data model and field definition will be determined during the ongoing phases. All the table definitions will be documented in the Data Dictionary document (found in *Appendix D* of this document).



7 Assumptions and Constraints

While the guiding principles establish the general values that the target architecture should consider, a number of assumptions were made about both the infrastructure and general direction for technology.

User Acceptance Test / System Acceptance Test Environment will be available for performing Usability testing, User Acceptance testing, Installation & Configuration testing and Performance Testing
General Architecture principles based on past experiences and Indusry Best practices & methodologies will be used in designing the solution
The basic TCP/IP (HTTP) protocol will be the only one used to access the application
The web browser will be the primary client used by employees and public users

Appendix A: Acronyms, Abbreviations, Terms and Definitions API	Application Program Interface

BO	Business Object
DAO	Data Access Object
DHTML	Dynamic Hypertext Markup Language
DMZ	De-Militarized Zone – Term for the portion of the network between the external Internet and the internal private network. The DMZ is protected from the outside by a Firewall.
DSB	Dynamic Screen Builder
EAR	Enterprise Archive
EDG	Eligibility Determination Group
EJB	Enterprise JavaBeans
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
LAN	Local Area Network – Communications network confined to the same physical building.
SMTP	Simple Mail Transport Protocol – Standard method of delivering internet email messages
J2EE	Java Enterprise Edition
JAR	Java Archive
JCA	Java Connector Architecture
JDBC	Java Database Connectivity
JRE	Java Runtime Environment
JSP	Java Server Pages
JVM	Java Virtual Machine
POJO	Plain Old Java Object
SQL	Structured Query Language
UML	Unified Modeling Language
WAR	Web Archive
XML	Extensible Markup Language
XSLT	Extensible Style Language Transformation

Appendix B: Products & Tools

The following software components will be utilized in the Online Screening Tool architecture. New versions of software may be released during the development of the system. The implementation of these new versions will be evaluated on an individual basis in determining if and when they will be implemented. Cross compatibility issues must be addressed before implementing any new versions of software products.

J2SDK	1.6_<<latest>>	http://java.sun.com	Java SDK for API
J2EE /Java EE 6	1.3	http://java.sun.com	Java Enterprise Editi for Enterprise service
Ant	1.6.5	http://www.apache.org	To Build and Deploy Development
Oracle Driver	10g	**http://www.oracle.com**	JDBC Driver to conn to SQL Server databa
Oracle 11g Client	10g	**http://www.oracle.com**	Oracle Server Client software
WinCVS <<Subversion instead>>	1.2	http://www.cvs.org	Version Control clier
CCD Framework	2.0	N/A	J2EE application Framework Generato for generating Cargo, Collections and DAC
JBoss	5 GA	http://www.jboss.org	Runtime for CCD Framework
Infrastructure Software/Tool	Version	Source	Description
Java JDK	1.4.2_08	http://java.sun.com	Java Runtime for Por and Application Serv
WebSphere Application Server	5.1.1	http://ibm.com	WebSphere as Application Server
IBM HTTP Webserver	1.3.28	http://www.ibm.com	Frond-end Web serve
Oracle 10g	10g	http://www.oracle.com	Oracle Server for database persistence
CVSNT	2.5.01	http://cvshome.org	Version Control tool Repository
VB Rules Generator	2.0	N/A	MS-Excel spreadshee XML Rules Convert

Appendix C: Configuration files

Below are some of key configuration files used in Online Screening Tool -

Application Configuration File

Below are three key application configuration files -

a) web.xml - The *Web application descriptor* provides the application server with information about the Web resources in the application.

b) application.xml - The application.xml file is the deployment descriptor for Enterprise Application Archives. The file is located in the META-INF subdirectory of the application archive

c) ejb-jar.xml - The EJB deployment descriptors contain structural and application assembly information for an enterprise bean. The ejb-jar.xml file is based on the deployment descriptors found in Sun Microsystems's ejb.jar.xml file.

Appendix D: Data Dictionary

Attached is data dictionary for the Online Screening Tool –

SCR_Dog_HOUSEHOLD	Household level information	SCR_ID	NUMB
SCR_Dog_HOUSEHOLD	Household level information	RENT_AMT	NUMB
SCR_Dog_HOUSEHOLD	Household level information	SW_UTILITIES	VARCI
SCR_GENERAL	HH statistical information	SCR_ID	NUMB
SCR_GENERAL	HH statistical information	CDE_USER	VARCI
SCR_GENERAL	HH statistical information	CDE_LOCATION	VARCI
SCR_GENERAL	HH statistical information	CDE_LANGUAGE	VARCI
SCR_INDIVIDUAL	Individual information	SCR_ID	NUMB
SCR_INDIVIDUAL	Individual information	INDV_ID	NUMB
SCR_INDIVIDUAL	Individual information	INDV_NAME	VARCI
SCR_INDIVIDUAL	Individual information	NUM_AGE	NUMB
SCR_INDIVIDUAL	Individual information	CDE_INDV_GENDER	VARCI