CIS 700B Master's Project Report

Eligibility Verification System

Submitted to the
Department of Computer Science
College of Computing Sciences
New Jersey Institute of Technology

In Partial Fulfillment of
The Requirements for the Degree of
Master of Science
By
Malathi Yellapragada

APPROVALS

Email: my33@njit.edu

Advisor: ____________________________________________
(Project Advisor Dr. James Geller)

Date Submitted: 12/12/05

Approved by: ________________________________________
(MS in CS Committee)

Date Approved: ________________________________
Abstract

This project focuses on New Jersey welfare services provided by New Jersey government agencies. In this project, the welfare services include Special Supplemental Nutrition Program for Women, Infants and Children (WIC), Work First New Jersey (WFNJ), NJ FamilyCare and NJ Fostercare. To be eligible for these programs, the applicants must be qualifying for eligibility requirements. In order to help in reducing the time and cost of applicants involved in manually searching for welfare services and entering information, and help them through the process of performing the eligibility verification efficiently, the NJ Welfare Eligibility Verification System was implemented.

In this system, a unified ontology was designed to store the key concepts and relationships of all the four welfare services. An online application form was designed for welfare applicants to fill out. Based on that, an eligibility verification engine was implemented to perform the eligibility verification. Furthermore, the input information of eligible applicants is sent to the simulated corresponding NJ government departments to process.

The system uses an ontology, and protégé, which is a very popular ontology editor. The system also uses JESS, which is the Java Expert System Shell. On top of that the system is implemented in Java and uses various Java technologies such as JSP, Servlets and Java Beans.
Abstract.......................................................................................................................... 2

Introduction ..................................................................................................................... 4

1 Problem Statement ................................................................................................. 5

2 Previous Work .......................................................................................................... 7

3 Knowledge of System Development ....................................................................... 8

3.1 Ontology .................................................................................................................... 9
3.2 Protégé 2000 ............................................................................................................ 9
3.3 JESS ........................................................................................................................... 10
3.4 JSP ............................................................................................................................ 10
3.5 JAVA BEANS ......................................................................................................... 10

4 Web Interface ......................................................................................................... 11

4.1 The Ontology of the NJ Welfare programs ............................................................. 11
4.2 The NJ Welfare Eligibility Verification Engine .................................................... 12
4.3 Simulated government websites ........................................................................... 19

5 Conclusion ............................................................................................................. 22

6 Future Work ............................................................................................................ 22


References ................................................................................................................. 25

Appendix A ................................................................................................................... 26

A.1 User Manual ............................................................................................................ 26
A.2 Database Scheme Design ...................................................................................... 28
A.3 Source Code .......................................................................................................... 31
Introduction

The explosion of the Internet has changed the way the world does business. Everyone is scrambling to keep up. Citizens or business entities are now facing too much information and web pages with too many hyperlinks pointing everywhere. The government has changed its traditional working style and implemented eGovernment projects. Many government agencies have put up websites to facilitate access to government welfare services, in order to make services user friendly for everyone. An applicant can visit each site and read the eligibility requirements. Searching the web to figure out eligibility is a time consuming and sometimes confusing process. Some sites even provide application forms to complete online. But, many sites do not provide such a form, leaving the applicant to travel to the agency.

To inform an applicant of their eligibility for government services may seem like a trivial task. One could simply hardcode the eligibility criteria into a program. However, any system like that would require a human to constantly monitor these government websites, and update the code as necessary when one of the eligibility criteria changes. This is where the difficulty lies. You need to manually change the software whenever there is a change in the welfare program or its eligibility rules. For this you have to constantly monitor the government websites for new changes. This will be very time consuming and tedious work.

Now since we are aware that everything is becoming web-based, one has to consider how this web-based technology can be made efficient and user friendly. If a person is new to computers and he/she is trying to fill out online forms, he/she will feel frustrated if he/she has to give similar information again and again. These input steps have to be
performed manually, in the right order. The reasons lie in the physical and logical
separation of web services. The Semantic web provides a common framework that
allows data to be shared and reused across applications, enterprises, and community
boundaries. "The Semantic web is an extension of the current web in which information
is given with well-defined meaning, better enabling computers and people to work in
cooperation." [1]

1 Problem Statement

This project helps in saving welfare applicants' time and efforts to find the welfare
programs they are eligible for, and also could help in improving the work efficiency of the
NJ government. The main functions and components of the NJ Welfare Eligibility
Verification System are as below:

First, a previous student [3] built a unified ontology to represent the main concepts and
their relationships in the domain of NJ Welfare Programs. Currently, the NJ Welfare
Eligibility Verification System only deals with four NJ welfare programs: Special
Supplemental Nutrition Program for Women, Infants and Children (WIC), Work First New
Jersey (WFNJ), NJ FamilyCare and NJ Fostercare. Therefore, the ontology stores
knowledge of the above four welfare programs, which includes entities and the eligibility
requirements of each NJ welfare program.

Second, provide one-stop-shop help for welfare applicants to conduct eligibility
verification. Put differently, the NJ Welfare Eligibility Verification System simulates the
real government staff to evaluate the applicants' qualifications. Welfare applicants only
need to fill out the online application form of the NJ Welfare Eligibility Verification

Malathi Yellapragada
System. The system will analyze the input information and compare the applicant’s information with the eligibility requirements stored in the back-end ontology. After this matching process is done, the system then lists which welfare programs the applicant is eligible for and the programs the applicant is not eligible for, and also lists the reasons why the applicant is not eligible for some welfare programs. Obviously, the eligibility verification results will help welfare applicants to pre-select their eligible welfare programs to apply to and thus save them time and effort.

Third, when the applicants first fill out the online application of the NJ Welfare Eligibility Verification System, their information will be saved in an Oracle Database. The applicant can come back again and retrieve his/her information later and he/she can also edit his/her information. They can even reapply to the programs, if any, that they were not eligible before.

Fourth, compose the services of the NJ Welfare Eligibility Verification System with the systems of corresponding NJ government agencies. This Verification System not only helps welfare applicants to conduct eligibility verification, but also helps the applicants to submit their information to the systems of the corresponding NJ government agencies to process. In other words, the NJ Welfare Eligibility Verification System will convert the format of the input data of eligible applicants to the data format of the corresponding government systems and then submit these data to the government agencies. As such, when welfare applicants submit their application forms via the NJ Welfare Eligibility Verification System, their information actually is also submitted to several (simulated) government systems. It is clear that the simultaneous submission to several agencies saves time and effort of welfare applicants.
The NJ Welfare Eligibility System is just a prototype system. Currently, the system has no relationship with NJ government agencies. Therefore, all the government websites involved in the project are simulated government websites.

2 Previous Work

Before I started this welfare project, several students had already participated and laid a foundation for my work. The previous work can be summarized as follows:

First of all, an ontology of the New Jersey welfare programs had been created. This ontology was built using Protégé-2000, which is a very popular tool for defining and organizing concepts and their relationships in a specific domain. In this ontology, the welfare programs Special Supplemental Nutrition Program for Women, Infants and Children (WIC), Work First New Jersey (WFNJ), NJ FamilyCare and NJ Fostercare were defined.

Secondly, the eligibility verification of the welfare programs Special Supplemental Nutrition Program for Women, Infants and Children (WIC), Work First New Jersey (WFNJ), NJ FamilyCare and NJ Fostercare had been implemented. When welfare applicants input and submit their information, the system informed the applicants whether or not they were eligible for the corresponding programs. The eligibility verification engine compares the applicants’ input information with the eligibility requirements stored in the back-end ontology. Unfortunately, in this implementation, the applicant information is not saved. That means there is no database which can store the information of the customers. Applicants have to re-type all his/her information again and
again. In other words, all the information was be lost once the applicant closes the window. Therefore, my work needed to create a Database where all information about the applicants and their eligibility rules can be saved. Every time the applicant logs in, his information is automatically populated in the form. I created the programs to establish JDBC connections to the Oracle database and save and reload the applicant's profile.

Thirdly, in the previous version of the NJ Welfare Eligibility Verification System, there is no login page functionality. Put differently, the previous system only provided the functionality of providing the eligibility rules to the applicants. Applicants couldn’t register by his/her username and password. Therefore, my work was to create a login page through which different users can enter the system with a particular username and password. This provides added security to the system. Through this, the system can better stop hackers from stealing personal information. Also, when already registered users login into the system, their information is automatically retrieved from the database.

Finally, the previous version of the NJ Welfare Eligibility Verification System did not provide logout page. After submitting their information, applicant’s eligibility rules will be displayed. To come out of that page, the applicant has to use browsers close button. This will be very inconvenient to the customers. Therefore, my work was to create a logout button.

3 Knowledge of System Development

The knowledge required for the system development in this project included the following aspects:
3.1 Ontology

An ontology is a specification of a conceptualization [1]. Ontology is a term originating from Philosophy, which refers to a systematic account of existence. In the field of Artificial Intelligence, ontology means an explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them. In the field of Computer Science, ontology represents the hierarchical structuring of knowledge about things by subcategorizing them according to their essential (or at least relevant and/or cognitive) qualities. Simply stated, an ontology is a representation of the main concepts and their relationship in a specific domain. Ontologies play a vital role in the emerging Semantic web. [8]

3.2 Protégé 2000

Protégé 2000 was developed by the Knowledge Modeling Group at Stanford University [6]. It is a platform which can be extended with graphical widgets for tables, or diagrams and animation components to access other knowledge-based systems and embedded applications. It is a library which other applications can use to access and display knowledge bases. The modules that use this tool can implement decision support systems, capture software requirements, populate databases, generate Java classes and UML diagrams, share and reuse domain models and access the Semantic web. This project involves downloading and installing Protégé software and applying it to model the knowledge required for the New Jersey Government Welfare Services.
3.3 JESS

Java Expert System Shell (JESS) is a program that uses a scripting language based on CLIPS. JESS is highly integrated with Java allowing you to call JESS API functions within Java code, or call Java functions within a JESS script file. [4]

Rule

A JESS rule is something like an if... then statement in a procedural language, but it is not used in a procedural way. While if... then statements are executed at a specific time and in a specific order, according to how the programmer writes those, JESS rules are executed whenever their if parts (their left-hand-sides or LHSs) are satisfied, given only that the rule engine is running. This makes JESS rules less deterministic than a typical procedural program. [3]

3.4 JSP

Java Server Pages (JSP) technology enables web developers and designers to rapidly develop and easily maintain, information-rich, dynamic web pages that leverage existing business systems. As part of the Java technology family, JSP technology enables rapid development of web-based applications that are platform independent. JSP technology separates the user interface from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content. This project is using JSP to implement the eligibility verification process for different applicants.

3.5 JAVA BEANS

JavaBeans technology is the component architecture for the Java 2 Platform, Standard Edition (J2SE). Components (JavaBeans) are reusable software programs that you can develop and assemble easily to create sophisticated applications. [5]
4 Web Interface

In this section, I will elaborate the design and implementation of the NJ Welfare Eligibility Verification System.

4.1 The Ontology of the NJ Welfare programs

The ontology of the NJ welfare programs is defined in a file “Rules.pprj”. It defines the entities existing in the welfare programs of WIC, NJWorkfirst, NJFamilycare and NJFoster, and also stores the eligibility requirements of all four welfare programs. The result of creating an ontology project in Protégé 2000 consists of three files: rules.pprj, rules.pont and rules.pins.

In Figure 1, we can see that all the concepts in the domain of the NJ welfare program are defined and organized as a hierarchical structure, which is shown in the left side of the ontology editor. In the right side of the ontology editor, all the necessary attributes of a specific entity are defined as slots.
4.2 The NJ Welfare Eligibility Verification Engine

The main functionality of the NJ Welfare Eligibility Verification Engine is to compare applicants’ input information with the eligibility requirements and rules stored in the welfare ontology and return the results to the applicants. Figure 2 is a snapshot of the homepage of the NJ Welfare Eligibility Verification System.
When activating the NJ Welfare Eligibility Verification Engine, new users must sign up before using the system and existing applicants must login into the system. Figure 3 shows a snapshot of the login page.
Figure 3: The Login page of NJ Welfare Verification System

When new users register themselves, following acknowledgement page shows up.

Figure 4: Thankyou Page of NJ Welfare Verification System
When a user enters invalid username and/or password, the following page shows up.

![Unauthorized page of NJ Welfare Verification System]

Figure 5: Unauthorized page of NJ Welfare Verification System

Once the new user logs into the system, he can access an online application form. This online application form includes the questions that will be needed for all four NJ welfare programs. Thus, by filling out this application form, welfare applicants have actually filled out four application forms simultaneously. Obviously, this unified online application form saves time and effort of welfare applicants. Figure 6a and 6b are the snapshots of this unified online application form.

When welfare applicants click the SAVE button, the NJ Welfare Eligibility Verification Engine will be activated to conduct the eligibility verification process.
Figure 6a: Online application form of the NJ Welfare Eligibility Verification System

Figure 6b: Online application form of the NJ Welfare Eligibility Verification System
Figure 7 is a snapshot of this comparison result for a welfare applicant. From Figure 7, we find that this applicant is eligible for the NJFoster program but he is not eligible for the NJWIC program and NJWorkfirst program, and the NJFamilycare program.

When an applicant logs into the system, his information is automatically populated. He may modify his information and his data will be updated in the database. Figure 8 shows a snapshot of retrieved data from the database.
Figure 8: The retrieved data of a welfare applicant
When the applicant clicks on logout, he/she is prompted to confirm his/her action with the following screen. Figure 9 shows a snapshot of the logout screen.

![Figure 9: The Logout screen of NJ Welfare Verification System](image)

### 4.3 Simulated government websites

Currently, since the welfare project is only a research project, and it has no relationship with the NJ government agencies, a previous student [3] has built four simulated government websites for experimental use in the welfare project. In this section, the NJFamilycare program is used as an example to briefly introduce the functionality of the simulated government websites.

The Homepage of NJFamilycare program is shown in Figure 10. From Figure 10, we can see that, this simulated government website covers the following contents:

- What is the NJFamilycare program about
• What does the NJFamilycare program cover
• What are the eligibility requirements for the NJFamilycare program
• Online application form

It is clear that even if this is a simulated government website, it covers all the necessary information for the welfare applicants to understand the NJFamilycare program. Furthermore, not only the introduction content of the welfare program but also the program’s eligibility requirements are not invented but copied from the real government website. Therefore, the composition between the NJ Welfare Eligibility Verification System and the four simulated websites has laid a foundation for the real system composition in the future.

Figure 10: The simulated government website of NJFamilycare program
Each simulated government website provides an online application form for welfare applicants to fill out. The application form is designed to include not only the contact information of welfare applicants, but also the questions concerning the eligibility requirements for a specific welfare program. For example, the application form of the NJFamilycare program is shown in Figure 11. Some eligibility requirements are overlapping among different welfare programs. Therefore, welfare applicants have to fill out some information repeatedly if they want to apply for several welfare programs.

At the server side of each simulated website, there is a JSP program running to get all the information input by welfare applicants and submit the information to the corresponding government agencies. However, the JSP program doesn’t perform the eligibility verification for applicants.
5 Conclusion

In the short term, this system will greatly benefit welfare customers. By using this system, applicants may conduct self-help eligibility verification. Furthermore, this system provides applicants a one-stop-shop eligibility verification help, which means applicants don’t need to go to different government agencies or search different government websites to find whether or not they are eligible for a specific welfare program; instead, this system automatically performs a series of eligibility verification processes for applicants and returns a list of eligible programs to them. To summarize, the system helps in saving applicants time and effort to search and find the existing programs that they are eligible for. The eligibility rules and policies are monitored, and automatically updated, whenever changes occur.

6 Future Work

This project will continue. Future work will be done by the next generation students. Some of the improvements that can be made are:

- The user interface needs to be designed more user-friendly: Some input fields should be divided into more fields like the fields NAME and DATE. It can be divided into first name, last name and middle name, and the field DATE could be divided into month, day and year. Moreover, in order to prevent users’ input errors and also improve user-friendliness, the field’s month, day and year could be provided as dropdown lists.
• Current system does not upload the data directly to the corresponding simulated government websites. The system should be enhanced to provide this functionality.

• The error tolerance is not very good - In the current user interface, the data format in the fields SSN and INCOME are too strict and thus lack error tolerance. For example, as to the field SSN, only when users input in the exact format “123-45-6789”, they can pass the data validation procedure. The better solution is to permit inputs such as “123456789” or “123-456789” to pass the data validation. The same problem also exists in the field INCOME. When one enters the income as “40,000”, he will receive a warning message of “income is not valid.” If he deletes the comma from the digit, he will pass the data validation procedure. So, the system should improve its error tolerance so as to provide users more convenience.

• Some sections that are not applicable to the applicant’s case are not disabled accordingly. Even though the current system provides a powerful function to automatically disable some sections that are not applicable to an applicant’s case, based on the analysis of the applicant’s input information, this function is still not implemented perfectly.

• In the future, the system should provide a PRINT function for the NJ Welfare Eligibility Verification System which would print all the user details in readable format.

7 Automatic Updating [2]

The rule processor is a file called rulebean.java. This file belongs in a directory called rules. The rules directory goes into your WEB-INF/classes folder. The ontology consists
of three files Rule.pprj, Rules.pont, Rules.pins. You can place these files wherever you like, however you must inform the rule processor, file called rulebean.java of its location.

To compile the rule bean, first change the path to the ontology in the rulebean.java file to point to the location where you placed the ontology. The variable to be changed is called PROJECT_FILE_NAME.

```java
private static final String PROJECT_FILE_NAME =
C:\tomcat\webapps\welfare1\ontology\Rules.pprj";
```

Then put the JESS API and Protégé API in WEB-INF/lib folder. Finally, run javac rules\rulebean.java.

The Rule Monitor and Updater consist of three files. The main file is rules.java. These files require 2 supporting files rulechecker.java and Utilities.java. They must all be in a directory called rules. To compile the Rule Monitor and Updater, first change the path to the ontology in the rulechecker.java file to point to the location where you placed the ontology. The variable to be changed is called PROJECT_FILE_NAME.

```java
private static final String PROJECT_FILE_NAME =
C:\tomcat\webapps\welfare1\ontology\Rules.pprj";
```

Then put the JESS API and Protégé API in lib folder. Finally, run javac rules\rules.java.

All the files are available in the directory called

afs/cad.njit.edu/research/Welfareservice/p/2/Malathi/edswork
References


[2] Ed Portscher Monitoring Eligibility Rules for Government Services, MS Report, NJIT, CS Dept,  

[3] Hongfang Shen NJ Welfare Eligibility Verification System, MS Report, NJIT, CS Dept,  


2005.

2005.

[7] JAVA API 1.4 - Sun Microsystems,  

Appendix A

A.1 User Manual

This section will describe the current system and set up the Welfare system.

Current system - This system integrated the two systems called Automatic Updating [2] and NJ Welfare Eligibility System [3] developed by previous students. This system uses the files called rulebean.java, rulechecker.java, rules.pprj from the Automatic Updating System and integrated with the NJ Welfare Eligibility System.

SetUp - Welfare system requires Tomcat as the web server. For instructions to setup and install Tomcat see http://jakarta.apache.org/tomcat/.

1. The home page of NJ Welfare Verification System is index.htm.
4. The result of creating an ontology project in Protégé 2000 consists of three files: Rules.pprj, Rules.pont and Rules.pins. Keep these in the ontology folder -- tomcat/webapps/welfare/ontology/ Rules.pprj
5. Keep the file rulebean.java, rulechecker.java in the folder WEB-INF/classes/rules/rulebean.java
6. WelfareDAO.java is the main file. Keep this file in the folder WEB-INF/classes/welfare/dao/WelfareDAO.class. This file validates the user and retrieves the user data.

7. LoginValidation.java is the file which runs when already registered user logs into the system again. Keep this file in the folder WEB-INF/classes/welfare/LoginValidation.class

8. Place all the DataBean.class files in the folder WEB-INF/classes/databean/

All the files are available in the directory called afs/cad.njit.edu/research/Welfareservice/p/2/Malathi/welfare

Welfare Application is available at http://web.njit.edu/welfare
A.2 Database Scheme Design

In this project, I adopted the concept of object relational table to store applicants’ input data. All these tables are available at [http://seer.njit.edu/isqlplus](http://seer.njit.edu/isqlplus)

```sql
create type address_type as object (
  street    varchar2(30),
  city      varchar2(20),
  state     varchar2(10),
  zip       varchar2(10),
  country   varchar2(20)
);

create type phones_type as object(
  home_phone  varchar2(20),
  work_phone  varchar2(20),
  cell_phone  varchar2(20)
);

create type contact_type as object ( 
  email varchar2(30),
  phones phones_type,
  address address_type
);

create type WIC_Beneficiary_type as object ( 
  ssn             varchar2(20),
  name               varchar2(30),
  sex            varchar2(30),
  immigrant_status  varchar2(60),
  date_enter_usa   varchar2(60),
  birth_date       varchar2(60),
  country_of_origin varchar2(20),
  country_of_citizenship varchar2(20),
  contact          contact_type
);

create type income_eligibility_type as object ( 
  family_size number(3, 0),
  income_max  number(10, 2)
);

create type doc_eligibility_type as object (
  NJ_residency_doc     varchar2(30),
  identity_doc         varchar2(30),
  nutritional_risk_doc varchar2(30)
);

create table WIC_beneficiary (  
  primary_key WIC_Beneficiary_type
  ...
)
```
CREATE TABLE tbl_users2 (
    USER_ID   VARCHAR2 (20)  NOT NULL,
    PASSWORD  VARCHAR2 (8)  NOT NULL,
    CONSTRAINT USER_PK
    PRIMARY KEY (USER_ID )) ;

CREATE TABLE care_preference_type (
    id varchar2(20) references tbl_users2(user_id),
    girl varchar2(1),
    boy varchar2(1),
    either varchar2(1),
    sibling varchar2(1),
    upto5 varchar2(1),
    upto10 varchar2(1),
    upto12 varchar2(1),
    above13 varchar2(1),
    physicalC varchar2(1),
    emotionalC varchar2(1),
    learningC varchar2(1),
    exposureC varchar2(1),
    hivC varchar2(1),
    behavioralC varchar2(1),
    medicalC varchar2(1),
    developmentalC varchar2(1)
)

CREATE TABLE foster_eligibility (
    id varchar2(20) references tbl_users2(user_id),
    WANT_FOSTER     CHAR(1),
    IS_HEALTHY      CHAR(1),
    IS_HOME_SAFE    CHAR(1),
    IS_FAMILY_INVOL CHAR(1),
    ATTEND_TRAINING CHAR(1)
)

CREATE TYPE work_status_type as object (
    has_assistance   varchar2(30),
    assistance_year     number(4, 2),
    current_work_status varchar2(60),
create type absent_parent_type as object (  
    absent_parent_name      varchar2(30),  
    absent_parent_ssn       varchar2(20),  
    absent_parent_address   varchar2(50),  
    absent_parent_DOB       varchar2(15),  
    absent_parent_employer  varchar2(50),  
    absent_parent_vehicle   varchar2(50),  
    absent_parent_plate_number varchar2(50)  
);  

create type teen_parent_type as object (  
    live_athome    varchar2(30),  
    finish_highschool varchar2(30),  
    attend_highschool varchar2(30)  
);  

create type drug_problem_type as object (  
    felony_drug_distribute varchar2(30),  
    drug_possesion         varchar2(30),  
    drug_treatment_program varchar2(30)  
);  

create table workfirst_beneficiary (  
    id                   varchar2(20) REFERENCES tbl_users2(USER_ID),  
    immigrant_status     varchar2(60),  
    date_enter_usa       date,  
    nj_residency         varchar2(30),  
    work_status          work_status_type,  
    has_children          varchar2(30),  
    absent_parent         absent_parent_type,  
    teen_parent           teen_parent_type,  
    has_drug_problem      varchar2(30),  
    drug_problem          drug_problem_type  
);
A.3 Source Code

Login.jsp

<html lang="en">
<head>

<LINK REL="stylesheet" href="/welfare/welfare.css"/>
<title>NJ Welfare Eligibility Verification System</title>

<script language="javascript">
    if ( window.opener && !window.opener.closed &&
        window.opener.myOpenWindows != null )
    {
        window.opener.sendRequestToOpener( location.href );
        window.close();
    }

    function init()
    {
        document.loginForm.j_username.focus();
    }

    function loginButtonClick()
    {
        if ( document.loginForm.j_username.value == "" ||
            document.loginForm.j_password.value == "" )
        {
            if ( document.loginForm.j_username.value == "" )
            {
                alert("User ID is required.");
                document.loginForm.j_username.focus();
            }
        }
        else
        {
            alert("Password is required.");
            document.loginForm.j_password.focus();
        }
        return (false);
    }

    function clickedNewUser()
    {
        //Url for SignUp
        if ( document.loginForm.j_username.value == "" ||
            document.loginForm.j_password.value == "" )
        {
            if ( document.loginForm.j_username.value == "" )
            {
                alert("User ID is required.");
            }
        }
    }
</script>
</head>
<body>
</body>
document.loginForm.j_username.focus();
}
else
{
    alert("Password is required.");
    document.loginForm.j_password.focus();
}
else
{
window.location.href='/welfare/SignUp.jsp?j_username='+document.loginForm.j_username.value+'&j_password='+document.loginForm.j_password.value;
}
</script>
</head>

<!-- onload element forces focus on first form element, remove onload for pages that do not contain form elements or an error will occur -->
<body leftmargin="0" marginheight="0" marginwidth="0" rightmargin="0" topmargin="0" onload="init();">

<!-- empty secondary navigation begins -->
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td colspan="3" class="dataHeadings" width="10%" nowrap>&nbsp;</td>
</tr>
</table>
<!-- empty secondary navigation ends -->

<!-- spacer table starts here, can be used for future nav -->
<table width="100%" height="30" border="0" cellspacing="0" cellpadding="0">
<tr>
<td colspan="3"&nbsp;></td>
</tr>
</table>
<!-- spacer table ends -->

<!-- content/body area begins, used for page content and specific app navigation -->
<table width="100%" cellspacing="0" cellpadding="0" border="0">
<tr>
<!-- left hand margin begins, used for app navigation -->
<td width="15%" valign="top" nowrap><![endif-->
<!--[if !IE]>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
Welcome to NJ Welfare Eligibility Verification System

Please log in below to access the application.

User ID:

Password:

Login
New User!! <a href="javascript:clickedNewUser();">SignUp</a>

<!-- content area content ends here --></td>
</tr>
</table>
</FORM>

<!-- content area content ends here --></td>
</tr>
</table>
</td>

<!-- middle content/app cell ends -->
<!-- right area begins, used for decoration, if not needed remove to gain more screen real estate, just be sure to change colspan from 3 to 2 in the top banner, secondary banner, and footer -->
<td width="15%" valign="top" nowrap>
<!-- right area content starts here --><!-- right area content ends here --></td>
<!-- right area margin ends -->
</tr>
</table>
<!-- content/body ends -->
<!-- footer begins -->

<!-- BEGIN FOOTER -->
<!-- BEGIN APP FOOTER -->
<!-- END APP FOOTER -->
<!-- END FOOTER -->
<!-- footer ends -->
</body>
</html>
<html lang="en">
<head>

<link rel="stylesheet" href="/welfare/welfare.css"/>
<title>NJ Welfare Eligibility Verification System</title>
<title>RailSync Command</title>
<script language="JavaScript">
var exit=true;
// Added this function to fix GNAT #4604
function KeepEm(){
    if (exit){
        whereto = '/command/';
        parent.backTo = whereto;
        window.location = parent.backTo;
    }
}
</script>
</head>
<body leftmargin="0" marginheight="0" marginwidth="0" rightmargin="0"
topmargin="0" onunload="KeepEm()">
<!-- top banner and basic site navigation starts here -->
</body>
</html>
<h1>UNAUTHORIZED ACCESS ATTEMPT</h1>

Invalid username and/or password, please try again.

You may &nbsp;&nbsp;<b><a href="/command/">Click Here</a></b> to login as a different user who has access to the requested page, or you may press your browser's "Back"
Logout.jsp

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><!-- InstanceBegin template="/Templates/njwv.dwt"
codeOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>NJ Welfare Verification System (Simulation)</title>
<!-- InstanceBeginEditable name="doctitle" -->
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" --><!-- InstanceEndEditable -->
</head>
<body>
<table width="1018" border="0" cellpadding="0" cellspacing="0">
<!--DWLayoutTable-->
<tr>
<td width="1018" height="26" valign="top" bgcolor="#9933FF" >&nbsp;</td>
</tr>
<tr>
<td height="81" valign="middle" background="../../img/bg_flower.gif">
<div align="center"><strong><font color="#000099" size="6">NJ Welfare Eligibility Verification System (Simulation)</font></strong></div></td>
</tr>
<tr>
<td height="40" valign="middle"
bgcolor="#9933FF"><a href="/index.jsp"><font color="#FFFFFF">Home</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="/overview.jsp"><font color="#FFFFFF">System Overview</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="/links.jsp"><font color="#FFFFFF">Simulated Gov. Websites</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="/edit.jsp"><font color="#FFFFFF">Edit</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="/contact.jsp"><font color="#FFFFFF">Contact us</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="/logout.jsp"><font color="#FFFFFF">Logout</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="/Login.jsp"><font color="#FFFFFF">Login</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="/apply.jsp"><font color="#FFFFFF">Online Application</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n
</font><p>&nbsp;</p>
```
SaveForm.jsp

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><!-- InstanceBegin template="/Templates/njwv.dwt" codeOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>NJ Welfare Verification</title>
</head>
<body>
<%! 
    String strNewUser = "Old";
%>
<%
    strNewUser = (String)session.getAttribute("userType");
    if(strNewUser == null)
        strNewUser = "Old";
    if(session.getAttribute("user") == null) {
        <jsp:forward page="Login.jsp"/>
    }
%>
<%
    String strUser = (String) session.getAttribute("user");
%>
boolean blnFormSaved = true;

welfare.dao.WelFareDAO welFareDAO = new welfare.dao.WelFareDAO();

databean.AddressType addressType = new databean.AddressType();
addressType.setStreet(request.getParameter("street"));
addressType.setCity(request.getParameter("city"));
addressType.setState(request.getParameter("state"));
addressType.setZIP(request.getParameter("zip"));
addressType.setCountry(request.getParameter("country"));

databean.PhoneType phoneType = new databean.PhoneType();
phoneType.setHome(request.getParameter("home_phone"));
phoneType.setWork(request.getParameter("work_phone"));
phoneType.setCell(request.getParameter("cell_phone"));

databean.ContactType contactType = new databean.ContactType();
contactType.setEMail(request.getParameter("email"));
contactType.setAddress(addressType);
contactType.setPhone(phoneType);

databean.BeneficiaryDataBean beneficiaryDB = new databean.BeneficiaryDataBean();
beneficiaryDB.setSSN(request.getParameter("ssn"));
beneficiaryDB.setName(request.getParameter("name"));
beneficiaryDB.setSex(request.getParameter("sex"));
beneficiaryDB.setImmigrantStatus(request.getParameter("immigrant_status"));
beneficiaryDB.setBirthDate(request.getParameter("birthdate"));
beneficiaryDB.setEnterUsaDate(request.getParameter("date_entry_usa"));
beneficiaryDB.setCountryOrigin(request.getParameter("country_origin"));
beneficiaryDB.setCountryCitigen(request.getParameter("citizen"));
beneficiaryDB.setContact(contactType);

databean.IncomeEligibilityDataBean incomeEligibilityDB = new databean.IncomeEligibilityDataBean();
double familySize = 0;
double familyIncome = 0;
try {
    familySize = Double.parseDouble(request.getParameter("family_size"));
} catch (Exception e) {}
try {
    familyIncome = Double.parseDouble(request.getParameter("income"));
} catch (Exception e) {}
incomeEligibilityDB.setFamilySize(familySize);
incomeEligibilityDB.setFamilyIncome(familyIncome);

databean.BenefitEligibilityDataBean benifitEligibilityDB = new databean.BenefitEligibilityDataBean();
benifitEligibilityDB.setBenifitEligibility(request.getParameter("current_benefit"));
databean.NJDocTypeDataBean njDocTypeDB = new databean.NJDocTypeDataBean();
njDocTypeDB.setNJDocType(request.getParameter("resident_doc"));

databean.IdentityDocDataBean identityDocDataBean = new databean.IdentityDocDataBean();
identityDocDataBean.setIdentityType(request.getParameter("id"));
identityDocDataBean.setIdentityNumber(request.getParameter("id_number"));

nutritionalDocDataBean.setNutritionalDoc(request.getParameter("nutrition_risk"));

databean.DocEligibilityTypeDataBean docEligibilityTypeDB = new databean.DocEligibilityTypeDataBean();
docEligibilityTypeDB.setNJDocTypeDataBean(njDocTypeDB);
docEligibilityTypeDB.setIdentityDocDataBean(identityDocDataBean);
docEligibilityTypeDB.setNutritionalDocDataBean(nutritionalDocDataBean);

String strNJResidency = request.getParameter("NJ_resident");
//System.out.println("strNJResidency --> " + strNJResidency);
String strHasInsurance = request.getParameter("insurance");

if(!welFareDAO.addUserBeneficiaryInfo(strUser,
    beneficiaryDB,
    incomeEligibilityDB,
    benifitEligibilityDB,
    docEligibilityTypeDB,
    strNJResidency,
    strHasInsurance,
    strNewUser
)
    blnFormSaved = false;

String strWantFoster = request.getParameter("foster_care");
String strIsHealthy  = request.getParameter("foster_care_health");
String strIsHomeSafe = request.getParameter("foster_care_home");
String strIsFamilyInvol = request.getParameter("foster_care_family_member");
String strAttendTraining = request.getParameter("foster_care_training");

if(!welFareDAO.createFosterEligibility(strUser,
    strWantFoster,
    strIsHealthy,
    strIsHomeSafe,
    strIsFamilyInvol,
    strAttendTraining,
strNewUser

blnFormSaved = false;

String strImmigrantStatus = request.getParameter("immigrant_status");
String dateEnterUSA = request.getParameter("date_entry_usa");

databean.WorkStatusDataBean workStatusDataBean = new databean.WorkStatusDataBean();
    workStatusDataBean.setHasAssistance(request.getParameter("receive_assistance"));
    workStatusDataBean.setAssistanceYear(request.getParameter("assistance_year"));
    workStatusDataBean.setCurrentWorkStatus(request.getParameter("work_status"));
    workStatusDataBean.setWorkDeferReason(request.getParameter("deferral_status"));

String strHasChildren = request.getParameter("children");

databean.AbsentParentDataBean absentParentDataBean = new databean.AbsentParentDataBean();
    absentParentDataBean.setAbsentParentName(request.getParameter("absent_parent_name"));
    absentParentDataBean.setAbsentParentSSN(request.getParameter("absent_parent_ssn"));
    absentParentDataBean.setAbsentParentAddress(request.getParameter("absent_parent_address"));
    absentParentDataBean.setAbsentParentDOB(request.getParameter("absent_parent_DOB"));
    absentParentDataBean.setAbsentParentEmployer(request.getParameter("absent_parent_employer"));
    absentParentDataBean.setAbsentParentVehicle(request.getParameter("absent_parent_moter_model"));
    absentParentDataBean.setAbsentParentLicenseNumber(request.getParameter("absent_parent_license_number"));


databean.TeenParentDataBean teenParentDataBean = new databean.TeenParentDataBean();
    teenParentDataBean.setLiveAtHome(request.getParameter("teen_athome"));
    teenParentDataBean.setFinishHighSchool(request.getParameter("finish_highschool"));
    teenParentDataBean.setAttendHighSchool(request.getParameter("attend_highschool"));

String strHasDrugProblem = request.getParameter("abuse_problem");

databean.DrugProblemDataBean drugProblemDataBean = new databean.DrugProblemDataBean();
drugProblemDataBean.setFelonyDrugDistribute(request.getParameter("felony_drug"));
drugProblemDataBean.setDrugPossesion(request.getParameter("drug_use"));
drugProblemDataBean.setDrugTreatmentProgram(request.getParameter("drug_use_free"));

if(!welFareDAO.createWorkFirstBeneficiary(strUser,
        strImmigrantStatus,
        dateEnterUSA,
        strNJResidency,
        workStatusDataBean,
        strHasChildren,
        absentParentDataBean,
        teenParentDataBean,
        strHasDrugProblem,
        drugProblemDataBean, strNewUser))

    blnFormSaved = false;

String strGirl = request.getParameter("foster_care_girl");
if(strGirl == null)
    strGirl = "";
String strBoy = request.getParameter("foster_care_boy");
if(strBoy == null)
    strBoy = "";
String strEither = request.getParameter("foster_care_either");
if(strEither == null)
    strEither = "";
String strSibling = request.getParameter("foster_care_sibling");
if(strSibling == null)
    strSibling = "";
String strUpto5 = request.getParameter("foster_care_age0_5");
if(strUpto5 == null)
    strUpto5 = "";
String strUpto10 = request.getParameter("foster_care_age5_10");
if(strUpto10 == null)
    strUpto10 = "";
String strUpto12 = request.getParameter("foster_care_age10_12");
if(strUpto12 == null)
    strUpto12 = "";
String strAbove13 = request.getParameter("foster_care_age13up");
if(strAbove13 == null)
    strAbove13 = "";
String strPhysical = request.getParameter("foster_care_physical");
if(strPhysical == null)
    strPhysical = "";
String strEmotional = request.getParameter("foster_care_emotional");
if(strEmotional == null)
    strEmotional = "";
String strLearning = request.getParameter("foster_care_learning");
if(strLearning == null)
    strLearning = "";
String strExposure = request.getParameter("foster_care_abuse");
if(strExposure == null)
    strExposure = "";
String strHIV = request.getParameter("foster_care_HIV");
if(strHIV == null)
    strHIV = "";
String strBehavioural = request.getParameter("foster_care_behavioral");
if(strBehavioural == null)
    strBehavioural = "";
String strMedical = request.getParameter("foster_care_medical");
if(strMedical == null)
    strMedical = "";
String strDevelopment = request.getParameter("foster_care_developmental");
if(strDevelopment == null)
    strDevelopment = "";
if(!weFareDAO.createFosterCarePreference(strUser,
    strGirl,
    strBoy,
    strEither,
    strSibling,
    strUpto5,
    strUpto10,
    strUpto12,
    strAbove13,
    strPhysical,
    strEmotional,
    strLearning,
    strExposure,
    strHIV,
    strBehavioural,
    strMedical,
    strDevelopment,
    strNewUser)) {
    blnFormSaved = false;
}
if(blnFormSaved) {
    response.sendRedirect(response.encodeRedirectURL("/welfare/NJ_WelfareVerify.jsp"));
} else {
    %>
    There is some problem in saving the form. Please try again.....
    <%
    }
SignUp.jsp

<%  
String strUser = request.getParameter("j_username");  
String strPassword = request.getParameter("j_password");  
welfare.dao.WelFareDAO welFareDAO = new welfare.dao.WelFareDAO();  
boolean blnUser = welFareDAO.addUser(strUser, strPassword);  
if(blnUser) {    
    session.setAttribute("user", strUser);    
    session.setAttribute("userType", "New");    
    response.sendRedirect(response.encodeRedirectURL("ThankYou.jsp"));  
} else {    
<%>
    <jsp:forward page="Login.jsp">    
    <jsp:param name="Message" value="Try with different User Name"/>    
    </jsp:forward>
<%
}  
%>

ThankYou.jsp

<html lang="en">  
<head>    
&lt;LINK REL="stylesheet" href="/welfare/welfare.css"/>&gt;    
<title>NJ Welfare Eligibility Verification System</title>  
</head>  
<body leftmargin="0" marginheight="0" marginwidth="0" rightmargin="0" topmargin="0" onload="init();">
<br>&nbsp;&nbsp;&nbsp;Thank you for choosing &lt;B&gt;&lt;FONT FACE=VERDANA SIZE=4&gt;"NJ Welfare Eligibility Verification System (Simulation)"&lt;/FONT&gt;&lt;/B&gt;
<br><br>
LoginValidation.java

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import welfare.dao.WelFareDAO;

public class LoginValidation extends HttpServlet{
    WelFareDAO welFareDAO;

    public void init() {
        welFareDAO = new WelFareDAO();
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
    {
        PrintWriter out    = response.getWriter();
        String strUser     = request.getParameter("j_username");
        String strPassword = request.getParameter("j_password");
        if(welFareDAO.validateUser(strUser, strPassword)) {
            HttpSession session = request.getSession(true);
            session.setAttribute("user", strUser);
            //getServletConfig().getServletContext().getRequestDispatcher("/apply.jsp").forward(request, response);
            response.sendRedirect(response.encodeRedirectURL("/welfare/NJ_WelfareVerify.jsp"));
        }
        else {
            getServletConfig().getServletContext().getRequestDispatcher("/login-error.jsp").forward(request, response);
        }
    }

    public static void main(String s[]) {
        (new LoginValidation()).init();
    }
}
apply.jsp

<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><!-- InstanceBegin template="/Templates/njwv.dwt" codeOutsideHTMLIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<!-- InstanceBeginEditable name="doctitle" -->
<title>NJ Welfare Verification</title>
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" -->
<!-- InstanceEndEditable -->
</head>

<%@ page import="java.util.*, databean.*, welfare.dao.*" %>

<script language="JavaScript">
function CheckDate(ValidDate) {
    var err = true;
    var frm = ValidDate;
    var DateValue=frm.value
    Month = DateValue.substr(0, 2)// month
    Slash1 = DateValue.substr(2, 1)// '/'
    Day = DateValue.substr(3, 2)// day
    Slash2 = DateValue.substr(5, 1)// '/'
    Year = DateValue.substr(6, 4)// year
    if (Month < 1 || Month > 12) {
        err = false
    }
    if (Slash1 != '/') {
        err = false
    }
    if (Day < 1 || Day > 31) {
        err = false
    }
    return err;
}
</script>
if (Slash2 != '/') {
    err = false
}

if (Year<1980 || Year>2100 || isNaN(Year)) {
    err = false
}

if (Month == 4 || Month == 6 || Month == 9 || Month == 11){
    if (Day == 31) {
        err=false
    }
}

if (Month == 2){
    var g=parseInt(Year/4)
    if (isNaN(g)) {
        err=false
    }
    if (Day > 29) {
        err=false
    }
    if (Day == 29 && ((Year/4) != parseInt(Year/4))) {
        err=false
    }
}

if (err == false) {
    alert('You have entered an invalid date, please re-enter.');
    return false
} //if (err == false)
else {
    return true
}

function DateCheck(date) {
    return CheckDate(date);
}
function validRequired(formField, fieldLabel) {
    var result = true;

    if (formField.value == "") {
        alert('Please enter a value for the "' + fieldLabel + '" field.');
        formField.focus();
        result = false;
    }
    return result;
}

function ValidateForm() {
    // Start ------>

    if(personal_info.ssn.value.length != 0) {
        if(!SSNValidation(personal_info.ssn.value)) {
            personal_info.ssn.focus();
            return false;
        }
    }

    if(personal_info.absent_parent_ssn.value.length != 0) {
        if(!SSNValidation(personal_info.absent_parent_ssn.value)) {
            personal_info.absent_parent_ssn.focus();
            return false;
        }
    }

    if(personal_info.birthdate.value.length != 0) {
        if(!DateCheck(personal_info.birthdate)) {
            personal_info.birthdate.focus();
            return false;
        }
    }

    if(personal_info.date_entry_usa.value.length != 0) {
        if(!DateCheck(personal_info.date_entry_usa)) {
            personal_info.date_entry_usa.focus();
            return false;
        }
    }

    if(personal_info.absent_parent_DOB.value.length != 0) {
        if(!DateCheck(personal_info.absent_parent_DOB)) {
            personal_info.absent_parent_DOB.focus();
            return false;
        }
    }

    if(!validRequired(personal_info.name, "Name"))
        return false;
    if(!validRequired(personal_info.birthdate, "Date of Birth"))

return false;
if(!validRequired(personal_info.family_size, "Family Size"))
return false;
if(!validRequired(personal_info.income, "Family Annual Income"))
return false;
if(!validRequired(personal_info.assistance_year, "Assistance Years"))
return false;
return true;
}

function SSNValidation(ssn) {
var matchArr = ssn.match(/^(\d{3})-[?]?\d{2}[-]?\d{4}$/);
var numDashes = ssn.split('-').length - 1;
var result = true;
if (matchArr == null || numDashes == 1)
{
alert('Invalid SSN. Must be 9 digits or in the form NNN-NN-NNNN.');
msg = "does not appear to be valid";
result = false;
}
else
if (parseInt(matchArr[1], 10) == 0)
{
alert("Invalid SSN: SSN's can't start with 000.");
msg = "does not appear to be valid";
result = false;
}
return result;
}

function validateZIP(field) {
var valid = "0123456789-";
var hyphencount = 0;
if (field.length != 5 && field.length != 10)
{
alert("Please enter your 5 digit or 5 digit+4 zip code.");
return false;
}
for (var i = 0; i < field.length; i++)
{
  temp = "" + field.substring(i, i + 1);
  if (temp == ".") hyphencount++;  
  if (valid.indexOf(temp) == "-1")
  {
alert("Invalid characters in your zip code. Please try again.");
return false;
}
if ((hyphencount > 1) || ((field.length==10) && ""+field.charAt(5)!="-"))
{
alert("The hyphen character should be used with a properly formatted 5 digit+four zip code, like '12345-6789'. Please try again.");
return false;
}
return true;
</SCRIPT>

<body>
<% if(session.getAttribute("user") == null) {
  
  <jsp:forward page="Login.jsp"/>

} %>

<% String strUser = (String) session.getAttribute("user");
WelfareDAO welFareDAO = new WelfareDAO();
UserBeneficiaryInfo userBeneficiaryInfo = welFareDAO.getUserBeneficiaryInfo(strUser);
IncomeEligibilityDataBean incomeEligibilityDataBean = userBeneficiaryInfo.getIncomeEligibilityDataBean();
BeneficiaryDataBean beneficiaryDataBean = userBeneficiaryInfo.getBeneficiaryDataBean();
WorkFirstBeneficiary workFirstBeneficiary = welFareDAO.getWorkFirstBeneficiary(strUser);
DrugProblemDataBean drugProblemDataBean = workFirstBeneficiary.getDrugProblemDataBean();
WorkStatusDataBean workStatusDataBean = workFirstBeneficiary.getWorkStatusDataBean();
AbsentParentDataBean absentParentDataBean = workFirstBeneficiary.getAbsentParentDataBean();
TeenParentDataBean teenParentDataBean = workFirstBeneficiary.getTeenParentDataBean();

String strLiveAtHome = teenParentDataBean.getLiveAtHome();
if(strLiveAtHome == null)
  strLiveAtHome = "";
String strFinishHighSchool = teenParentDataBean.getFinishHighSchool();
if(strFinishHighSchool == null)
    strFinishHighSchool = "";

String strAttendHighSchool = teenParentDataBean.getAttendHighSchool();
if(strAttendHighSchool == null)
    strAttendHighSchool = "";

String strAbsentParentName = absentParentDataBean.getAbsentParentName();
if(strAbsentParentName == null)
    strAbsentParentName = "";

String strAbsentParentPlateNumber =
absentParentDataBean.getAbsentParentPlateNumber();
if(strAbsentParentPlateNumber == null)
    strAbsentParentPlateNumber = "";

String strAbsentParentVehicle = absentParentDataBean.getAbsentParentVehicle();
if(strAbsentParentVehicle == null)
    strAbsentParentVehicle = "";

String strAbsentParentEmployer =
absentParentDataBean.getAbsentParentEmployer();
if(strAbsentParentEmployer == null)
    strAbsentParentEmployer = "";

String strAbsentParentDOB = absentParentDataBean.getAbsentParentDOB();
if(strAbsentParentDOB == null)
    strAbsentParentDOB = "";

String strAbsentParentAddress = absentParentDataBean.getAbsentParentAddress();
if(strAbsentParentAddress == null)
    strAbsentParentAddress = "";

String strAbsentParentSSN = absentParentDataBean.getAbsentParentSSN();
if(strAbsentParentSSN == null)
    strAbsentParentSSN = "";

String strNJResidency = workFirstBeneficiary.getNJResidency();
if(strNJResidency == null)
    strNJResidency = "y";

String strHasChildren = workFirstBeneficiary.getHasChildren();
if(strHasChildren == null)
    strHasChildren = "y";

String strFamilySize = String.valueOf(incomeEligibilityDataBean.getFamilySize());
if(strFamilySize == null)
    strFamilySize = "1";
String strFamilyIncome =
String.valueOf(incomeEligibilityDataBean.getFamilyIncome());
if(strFamilyIncome == null)
    strFamilyIncome = "0";

String strDrugPossesion = drugProblemDataBean.getDrugPossesion();
if(strDrugPossesion == null)
    strDrugPossesion = "August 22, 1996";

String strHasDrugProblem = workFirstBeneficiary.getHasDrugProblem();
if(strHasDrugProblem == null)
    strHasDrugProblem = "August 22, 1996";

String strDrugTreatmentProgram =
drugProblemDataBean.getDrugTreatmentProgram();
if(strDrugTreatmentProgram == null)
    strDrugTreatmentProgram = "61";

String strCountryCitizen = beneficiaryDataBean.getCountryCitigen();
if(strCountryCitizen == null)
    strCountryCitizen = "U.S. Citizen";

String strName = beneficiaryDataBean.getName();
if(strName == null)
    strName = "";

String strAssistanceYear = workStatusDataBean.getAssistanceYear();
if(strAssistanceYear == null)
    strAssistanceYear = "";

String strImmigrantStatus = beneficiaryDataBean.getImmigrantStatus();
if(strImmigrantStatus == null)
    strImmigrantStatus = "permanent resident";

String strBirthDate = beneficiaryDataBean.getBirthDate();
if(strBirthDate == null)
    strBirthDate = "";

String strSSN = beneficiaryDataBean.getSSN();
if(strSSN == null)
    strSSN = "";

String strCell = "";
try {
    strCell = beneficiaryDataBean.getCell();
} catch (Exception e ) {
    strCell="";
}
if(strCell == null)
    strCell = "";
String strSex = beneficiaryDataBean.getSex();
if(strSex == null)
    strSex = "F";

String strWork = "";
try {
    strWork = beneficiaryDataBean.getWork();
} catch (Exception e)
    { strWork = "";}
if(strWork == null)
    strWork = "";

String strHome = "";
try{
    strHome = beneficiaryDataBean.getHome();
} catch (Exception e)
    {strHome = "";}
if(strHome == null)
    strHome = "";

String strCountry = "";
try{
    strCountry = beneficiaryDataBean.getCountry();
} catch (Exception e){strCountry ="";}
if(strCountry == null)
    strCountry = "USA";

String strZIP = "";
try {
    strZIP = beneficiaryDataBean.getZIP();
} catch (Exception e){strZIP = "";}
if(strZIP == null)
    strZIP = "";

String strState = "";
try{
    strState = beneficiaryDataBean.getState();
} catch (Exception e){ strState ="";}
if(strState == null)
    strState = "";

String strCity = "";
try{
    strCity = beneficiaryDataBean.getCity();
} catch (Exception e){ strCity ="";}
if(strCity == null)
    strCity = "";

String strStreet = "";
try{
strStreet = beneficiaryDataBean.getStreet();
}catch (Exception e) {strStreet ="";}
if(strStreet == null)
        strStreet = "";

String strEMail ="";
try {
strEMail = beneficiaryDataBean.getEMail();
}catch (Exception e) {strEMail="";}
if(strEMail == null)
        strEMail = "";

String strCountryOrigin = "";
try{
strCountryOrigin = beneficiaryDataBean.getCountryOrigin();
}catch(Exception e) {strCountryOrigin = "";}
if(strCountryOrigin == null)
        strCountryOrigin = "";

String strEnterUsaDate ="";
try{
 strEnterUsaDate = beneficiaryDataBean.getEnterUsaDate();
} catch(Exception e) { strEnterUsaDate ="";}
if(strEnterUsaDate == null)
        strEnterUsaDate = "";

%>
<table width="1018" border="0" cellpadding="0" cellspacing="0">
        <!--DWLayoutTable-->
        <tr>
            <td width="1018" height="26" valign="top" bgcolor="#9933FF" >
                <!--DWLayoutEmptyCell-->
                &nbsp;</td>
        </tr>
        <tr>
            <td height="81" valign="middle" background="img/bg_flower.gif">
                <div align="center"><strong><font color="#000099" size="6">NJ Welfare Eligibility Verification System (Simulation)</font></strong></div>
            </td>
        </tr>
        <tr>
            <td height="40" valign="middle">
                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n

</td>
</tr>
</table>
Information Required

Notes:

- The following form should be filled out in the name of beneficiary;

- If the beneficiary is a child under 18 years old, the contact information here refers to the child's parent's contact information;

- Family size includes children under 21 years old and their parents. Grandparents, caretakers and guardians are not counted in the family size. Their income is also not counted as family income.

Beneficiary Information:

<form method="POST" action="SaveForm.jsp" name="personal_info">
<table border="0" width="60%">
<tr>
<td width="24%">Name</td>
<td><input name="name" size="40" value = ""><br>
</td>
</tr>
<tr>
<td width="24%">Gender</td>
<td><input type="radio" value="F" checked name="sex"> Female
</td>
</tr>
<br>
</table>
</form>
| **SSN** | <input name="ssn" value="<%= strSSN%>" tabindex="3">
| --- | e.g. 123-45-6789 |
| **Date of Birth** | <input name="birthdate" value="<%= strBirthDate%>" tabindex="5">
| --- | (MM/DD/YYYY) |
| **Country of Origin** | <input name="country_origin" value="USA" tabindex="6">
| **Are you a US Citizen** | <input name="citizen" type="radio" value="U.S. Citizen" checked>
| --- | Yes |
| --- | No |

*if(strCountryCitizen.equals("U.S. Citizen")) {}*
Yes
<input type="radio" name="citizen" value="n" checked>
No</font></td>

</tr>
</table>

<tr>
<td nowrap><font face="Times New Roman, Times, serif">Immigrant status</font></td>
<td nowrap><select name="immigrant_status">
<%}
<%}%>
</select></td>
</tr>
<tr>
<td nowrap><font face="Times New Roman, Times, serif">The date when the applicant entered the U.S.A.</font></td>
<td nowrap><font face="Times New Roman, Times, serif">
<input name="date_entry_usa" tabindex="7" value="\" + strEnterUsaDate + "/MM/DD/YYYY"/>
</font></td>
</tr>
Is the applicant a NJ resident?<br>

```html
<tr><td width="24%" nowrap><font face="Times New Roman, Times, serif">Is the applicant a NJ resident?</font></td><td width="76%" nowrap><font face="Times New Roman, Times, serif">
  <% if(strNJResidency.equals("y")) {
    <input name="NJ_resident" type="radio" value="y" checked> Yes
    <input type="radio" name="NJ_resident" value="n"> No</input>
  } else {
    <input name="NJ_resident" type="radio" value="y">
    Yes <input type="radio" name="NJ_resident" value="n" checked> No</input>
  }
  %>
</td></tr>
</table>

Contact Information:<br>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| Street Address| <input name="street" size="41" tabindex="8" value = "<%= strStreet%>" />
| City          | <input name="city" tabindex="9" value = "<%= strCity%>" />
| State         | <input name="state" size="6" value="NJ" tabindex="10" value = "<%= strState%>" />
| ZIP           | <input name="zip" tabindex="11" maxlength="5" value = "<%= strZIP%>" />
```
<table>
<thead>
<tr>
<th>Country</th>
<th>Input Country</th>
<th>Home Phone</th>
<th>Input Home Phone</th>
<th>Work Phone</th>
<th>Input Work Phone</th>
<th>Cell Phone</th>
<th>Input Cell Phone</th>
<th>Email</th>
<th>Input Email</th>
</tr>
</thead>
</table>
|         | <input name="country" size="22" tabindex="12" value=""/>
|         | <input name="home_phone" size="30" tabindex="13" value=""/>
|         | <input name="work_phone" size="30" tabindex="13" value=""/>
|         | <input name="cell_phone" size="30" tabindex="13" value=""/>
|         | <input name="email" size="37" tabindex="14" value=""/>

**Family Income:**

<table>
<thead>
<tr>
<th>Family Size</th>
<th>Input Family Size</th>
</tr>
</thead>
</table>
|             | <input name="family_size" size="8" value=""/>

*Family Size:*
<table>
<thead>
<tr>
<th>Family Annual Income</th>
<th>&lt;input name=&quot;income&quot; style=&quot;LEFT: 0px; TOP: 9px&quot; tabindex=&quot;16&quot; value=&quot;&quot;&gt;&lt;%= strFamilyIncome%&gt;</th>
</tr>
</thead>
</table>

Document Information:
- Please select one ID certificate that the beneficiary has.
  - <select size="1" name="id" tabindex="1">None</select> Passport Birth Certificate Driver License
- Document Number: <input type="text" name="id_number" size="20" tabindex="2">

Please select a document that the beneficiary has to prove his NJ residency.
- <select size="1" name="resident_doc" tabindex="3">None NJ Driver License Any Bill with Address NJ Library Card</select>

Please select a document that the beneficiary has to prove he is in nutritional risk.
- <select size="1" name="nutrition_risk" tabindex="4">None Diet History Report Blood Test Report Height/Weight Report Health History Report Other_Doc</select>

Please select any benefits the beneficiary has currently. (several may be selected)
- <select size="4" name="current_benefit" multiple tabindex="5">None Medicaid</select>
<option value="Foodstamps">Foodstamps</option>
<option value="Aid_For_Dependent_Children">Aid for Dependent Children</option>
</select>

<p align="left"></p>
<p align="left"><font color="#000089" size="5"><strong>Insurance Information:</strong></font></p>
<table width="75%" border="0">
<tr><td width="68%" nowrap><font face="Times New Roman, Times, serif">Did this applicant have health insurance in the past 6 months?</font></td><td width="32%" nowrap><input name="insurance" type="radio" value="y">Yes<input name="insurance" type="radio" value="n" checked>No</td></tr>
</table>
<p align="left"><b><font size="5" color="#000080">Work Status:</font></b></p>
<table border="0" width="60%">
<tr><td width="25%" nowrap><font face="Times New Roman, Times, serif">Has the applicant received WFNJ assistance before?</font></td><td width="76%" nowrap><input name="receive_assistance" type="radio" value="y" checked>Yes<input type="radio" name="receive_assistance" value="n">No</td></tr>
<tr><td width="25%" nowrap><font face="Times New Roman, Times, serif">If yes, how many years in total has the applicant received the assistance? (Note: if the assistance was received before age 18, the months of receipt are not counted.)</font></td><td width="76%" nowrap><input name="assistance_year" tabindex="11" maxlength="5" size="8" value = "<%= strAssistanceYear%>" years</td></tr>
<tr><td width="25%" nowrap><font face="Times New Roman, Times, serif">The applicant's current work status:</font></td><td width="76%" nowrap><select name="work_status">
<option value="employment" selected>Employment</option>
</select>
<option value="support_work">Supported Work</option>
<option value="community_work">Community Work Experience</option>
<option value="alternative_work">Alternative Work Experience</option>
<option value="job_training">On the Job Training</option>
<option value="job_search">Job Search/Job Readiness</option>
<option value="vocational_education">Vocational Education</option>
<option value="skill_training">Job Skills Training Related to Employment</option>
<option value="child_care">Provision of Child Care</option>
<option value="community_service">Community Service</option>
<option value="employment_education">Education Directly Related to Employment</option>
<option value="high_school">High School or GED Program</option>
<option value="college_study">College Work Study</option>
<option value="not_work">Unable to Work</option>
</select>
</td>
</tr>
<tr>
<td width="25%" nowrap><font face="Times New Roman, Times, serif">If the applicant has a temporary work deferral, select a deferral reason:</font></td>
<td>&nbsp;</td>
</tr>
<tr>
<td><font face="Times New Roman, Times, serif">
<select name="deferral_status" >
<option value="age60" selected>persons age 60 or older</option>
<option value="pregancy">women who have the seventh month of pregnancy</option>
<option value="child_care">parents or relatives who are taking care of a child under 12 weeks of age</option>
<option value="care_taker">sole caretaker of a severely disabled/ill dependent family member</option>
<option value="disability">persons who are physically or mentally impaired</option>
<option value="family_violence">victims of family/domestic violence</option>
<option value="none">Not Applicable</option>
</select>
</font></td>
<td>&nbsp;</td>
</tr>
</table>
<p><font color="#000089" size="5"><strong>Dependent Children</strong></font></p>
<table width="75%" border="0">
<tr>
<td width="68%" nowrap><font face="Times New Roman, Times, serif">Does the applicant have dependent children?</font></td>
</tr>
</table>
<td width="32%" nowrap><blockquote>
<p><font face="Times New Roman, Times, serif">
<% 
if(strHasChildren.equals("y")) {

<input name="children" type="radio" value="y" checked > Yes
<input name="children" type="radio" value="n" > No</font></p>
<%
}
else {

<input name="children" type="radio" value="y" > Yes
<input name="children" type="radio" value="n" checked > No</font></p>
<%
}

</blockquote></td>
</tr>
<tr>
<td width="25%" align="left" valign="top" nowrap><font face="Times New Roman, Times, serif"><br>
If yes, please provide the following information about the absent parent?<br>
</font></td>
<td width="76%" nowrap><blockquote>
<p><font face="Times New Roman, Times, serif"><br>
Name: <input name="absent_parent_name" tabindex="11" size="40" value = "<%= strAbsentParentName%>">
<br>
SSN: &nbsp;
<input name="absent_parent_ssn" tabindex="11" size="20" value = "<%= strAbsentParentSSN%>">
<br>
Address: <input name="absent_parent_address" tabindex="11" size="40" value = "<%= strAbsentParentAddress%>">
<br>
Date of birth: <input name="absent_parent_DOB" tabindex="11" size="20" value = "<%= strAbsentParentDOB%>" (MM/DD/YYYY)
<br>
Employer: <input name="absent_parent_employer" tabindex="11" size="40" value = "<%= strAbsentParentEmployer%>">
<br>
</font></p>
</blockquote>
make and model of motor vehicle:
<input name="absent_parent_moter_model" tabindex="11" size="20"
value=""% strAbsentParentVehicle">
<br>
license plate number:
<input name="absent_parent_license_number" tabindex="11" size="20"
value=""% strAbsentParentPlateNumber">
</font></p></blockquote></td></tr></table><p><font color="#000089" size="5" face="Times New Roman, Times," serif"><strong>For Teen Parents:<br></strong></font><table border=1><tr><td width="68%" nowrap><font face="Times New Roman, Times," serif">Does the applicant live at home with parents or<br>with a responsible adult relative?</font><font face="Times New Roman, Times," serif"> <br></font></td><td width="32%" nowrap><font face="Times New Roman, Times," serif"><% if(strLiveAtHome.equals("y")) {
<input name="teen_athome" type="radio" value="y" checked>
Yes
<input name="teen_athome" type="radio" value="n" />
No</font></td><%
else {%
<input name="teen_athome" type="radio" value="y" 
Yes
<input name="teen_athome" type="radio" value="n" checked>
No</font></td><%
} %></tr><tr><td width="68%" nowrap><font face="Times New Roman, Times," serif">Has the applicant finished high school?</font><font face="Times New Roman, Times," serif><br></font></td><td width="32%" nowrap><font face="Times New Roman, Times," serif"><% if(strFinishHighSchool.equals("y")) {
<input name="teen_athome" type="radio" value="y" checked>
Yes
<input name="teen_athome" type="radio" value="n" />
No</font></td><%
} %></tr></table>
%>
<input name="finish_highschool" type="radio" value="y" checked >
Yes
<input name="finish_highschool" type="radio" value="n" >
No</font></td>
<%
} else {
%
<input name="finish_highschool" type="radio" value="y" >
Yes
<input name="finish_highschool" type="radio" value="n" checked> No</font></td>
<%
%
</tr>
</tr>
<td width="68%" nowrap><font face="Times New Roman, Times, serif">If no, is the applicant attending a high school or GED program now?</font></td>
<td width="32%" nowrap><font face="Times New Roman, Times, serif">
<%
if(strAttendHighSchool.equals("y")) {
<%
<input name="attend_highschool" type="radio" value="y" checked >
Yes
<input name="attend_highschool" type="radio" value="n" >
No</font></td>
<%
} else {
%
<input name="attend_highschool" type="radio" value="y" >
Yes
<input name="attend_highschool" type="radio" value="n" checked > No</font></td>
<%
%
</tr>
</table>
<p><font color="#000089" size="5" face="Times New Roman, Times, serif"><strong>For applicants who have substance abuse problems : </strong></font></p>
<table border=1>
<tr>
<td width="68%" nowrap><font face="Times New Roman, Times, serif">Does the applicant have problems with drugs or alcohol?</font></td>
<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>If yes, was the applicant convicted of felony drug distribution on or after Aug. 22, 1996?</strong></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Does the applicant have a conviction of drug possession, or use only, of a controlled substance on or after Aug. 22, 1996?</strong></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
If yes, was the applicant enrolled in and completed a licensed residential drug treatment program, test at the end of the program and has remained drug free for 60 days?

If strDrugTreatmentProgram.equals("61") {
    <input name="drug_use_free" type="radio" value="61" checked> Yes
    <input name="drug_use_free" type="radio" value="1" > No
} else {
    <input name="drug_use_free" type="radio" value="61"> Yes
    <input name="drug_use_free" type="radio" value="1" checked> No
}

Foster Care Information:

Does the applicant want to be a foster care parent?

If yes, please fill in the following information:
<table>
<thead>
<tr>
<th>Question</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the applicant in good health?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Does the applicant's home meet basic life and safety standard?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Does everybody in the applicant's family agree to the commitment of providing foster care for a child?</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Is the applicant willing to attend pre-servicing training?</td>
<td>Yes/No</td>
</tr>
</tbody>
</table>

The applicant's Foster Care Preference:
- **Girls**
There is a particular need for foster homes for children with special challenges. Would the applicant consider fostering a child with any of the following needs?

<table>
<thead>
<tr>
<th>Boys</th>
<th>Either</th>
<th>Sibling Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-5</td>
<td>5-10</td>
<td>10-12</td>
</tr>
<tr>
<td>13 and up</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Physical
- Emotional
<input type="checkbox" name="foster_care_learning" value="y"> Learning

<input type="checkbox" name="foster_care_abuse" value="y"> Exposure to Substance Abuse

<input type="checkbox" name="foster_care_HIV" value="y"> HIV Positive

<input type="checkbox" name="foster_care_behavioral" value="y"> Behavioral

<input type="checkbox" name="foster_care_medical" value="y"> Medical

<input type="checkbox" name="foster_care_developmental" value="y"> Developmental

<form>
  <input type="submit" value="Save" name="Submit" onclick = "return ValidateForm();">
  <input type="reset" value="Reset" name="Reset">
</form>

NJ_WelfareVerify.jsp

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>NJ Welfare Verification</title>
</head>
<body>
</body>
</html>
<%@ page import="java.util.*, databean.*, welfare.dao.*" %>

<body>
  <%
    if(session.getAttribute("user") == null) {
      <jsp:forward page="Login.jsp"/>
    
  }
  try {
    <table width="1018" border="0" cellpadding="0" cellspacing="0">
      <!--DWLayoutTable-->
      <tr>
        <td width="1018" height="26" valign="top" bgcolor="#9933FF" >&nbsp;</td>
      </tr>
      <tr>
        <td height="81" valign="middle" background="img/bg_flower.gif">
          <div align="center"><strong><font color="#000099" size="6">NJ Welfare Eligibility Verification System (Simulation)</font></strong></div>
        </td>
      </tr>
      <tr>
        <td height="40" valign="middle" bgcolor="#9933FF">
          <a href="index.htm"><font color="#FFFFFF">Home</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
          <strong><a href="overview.htm"><font color="#FFFFFF">System Overview</font></a>&nbsp;&nbsp;</strong>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
          <a href="links.htm"><font color="#FFFFFF"><strong>Simulated Gov. Websites</strong></font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
          <a href="apply.jsp"><font color="#FFFFFF">Online Application</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
          <a href="contact.htm"><font color="#FFFFFF">Contact us</font></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
          <a href="Logout.jsp"><font color="#FFFFFF">LogOut</font></a>
        </td>
      </tr>
    </table>
  </body>
<tr><td height="835" valign="top" background="img/bg_flower.gif" ></p>&nbsp;</td></tr>

<!-- The following code calls the rule processing bean and displays its results -->

<jsp:useBean id="bean" class="rules.rulebean" scope="request"/>
</jsp:useBean>

<%@ page import="java.util.Vector" %>

//bean.setName("Ravi");name = beneficiaryDataBean.getName();
//bean.setFamily_size("4");family_size = String.valueOf(incomeEligibilityDataBean.getFamilySize());
//bean.setIncome("900");family_income = String.valueOf(incomeEligibilityDataBean.getFamilyIncome());
//felony_drug_distribute = drugProblemDataBean.getFelonyDrugDistribute();
//bean.setFelony_drug("y");drug_possession = drugProblemDataBean.getDrugPossession();
//bean.setDrug_use_free("70");drug_treatment_program = drugProblemDataBean.getDrugTreatmentProgram();
//bean.setDrug_use("y");has_drug_problem = workFirstBeneficiary.getHasDrugProblem();
//bean.setCitizen("U.S. citizen");citizenship = beneficiaryDataBean.getCountryCitigen();

String strUser = (String) session.getAttribute("user");
WelFareDAO welFareDAO = new WelFareDAO();
UserBeneficiaryInfo userBeneficiaryInfo = welFareDAO.getUserBeneficiaryInfo(strUser);
IncomeEligibilityDataBean incomeEligibilityDataBean = userBeneficiaryInfo.getIncomeEligibilityDataBean();
BeneficiaryDataBean beneficiaryDataBean = userBeneficiaryInfo.getBeneficiaryDataBean();
WorkFirstBeneficiary workFirstBeneficiary = welFareDAO.getWorkFirstBeneficiary(strUser);
DrugProblemDataBean drugProblemDataBean = workFirstBeneficiary.getDrugProblemDataBean();
WorkStatusDataBean workStatusDataBean = workFirst Beneficiary.getWorkStatusDataBean();

bean.setFileName(application.getRealPath("ontology/Rules.pprj"));
bean.setFamily_size(String.valueOf(incomeEligibilityDataBean.getFamilySize()));
bean.setIncome(String.valueOf(incomeEligibilityDataBean.getFamilyIncome()));
bean.setFelony_drug(String.valueOf(incomeEligibilityDataBean.getFelonyDrugDistribute()));
bean.setDrug_use(workFirstBeneficiary.getDrugPossession());
bean.setDrug_use_free(workFirstBeneficiary.getDrugTreatmentProgram());
bean.setDrug_use(workFirstBeneficiary.getHasDrugProblem());
bean.setCitizen(workFirstBeneficiary.getCountryCitigen());
bean.setName(beneficiaryDataBean.getName());
bean.setAssistance_year(workStatusDataBean.getAssistanceYear());
bean.setImmigrant_status(beneficiaryDataBean.getImmigrantStatus());
bean.setBirthdate(beneficiaryDataBean.getBirthDate());

Vector result = bean.queryeligibility();

for (int i=0;i<result.size();i++)
{
    out.println(result.elementAt(i));
}
catch(Exception e) {
    e.printStackTrace();
}%>
</tr>
</table>
</body>
</html>

WelFareDAO.java

package welfare.dao;

import java.sql.*;
import databean.*;

public class WelFareDAO {

    Connection con;
    Statement stmt;

    public WelFareDAO() {
        try {
            oracle.jdbc.driver.OracleDriver driver = new oracle.jdbc.driver.OracleDriver();
            //con = DriverManager.getConnection("jdbc:oracle:thin:@172.19.74.40:1521:srfweb", "srfadmin ", "srf4dm1n");
            con = DriverManager.getConnection("jdbc:oracle:thin:@seer.njit.edu:1521:research", "welfare", "farewell");

            System.out.println("Connection Established");
            stmt = con.createStatement();
        }catch(Exception e) {
public boolean validateUser(String strUser, String strPassword) {
    boolean blnValid = false;
    String strError = "";
    String strSQL = "SELECT * FROM tbl_users2 "
                   + "WHERE USER_ID = " + strUser + 
                   + "AND PASSWORD = " + strPassword + ";
    try {
        ResultSet rs = stmt.executeQuery(strSQL);
        if (rs.next()) {
            blnValid = true;
        }
    } catch (Exception exp) {
        strError = "NJW-001: An error has occurred in validateUser() in WelFareDAO"
                   + exp.getMessage() + "n";
        strError += exp.getMessage() + "n";
        strError += "SQL: " + strSQL;
        System.out.println(strError);
    }
    return blnValid;
}

public boolean addUser(String strUser, String strPassword) {
    boolean blnValid = true;
    String strError = "";
    String strSQL = "SELECT * FROM tbl_users2 "
                   + "WHERE USER_ID = " + strUser + ";
    try {
        ResultSet rs = stmt.executeQuery(strSQL);
        if (rs.next()) {
            blnValid = false;
        } else {
            strSQL = "INSERT INTO tbl_users2 VALUES(" + strUser + "]", " + strPassword + "]");
            int i = stmt.executeUpdate(strSQL);
            //                con.commit();
            if (i != 1)
                blnValid = false;
        }
    } catch (Exception exp) {
        strError = "NJW-002: An error has occurred in addUser() in WelFareDAO"
                   + exp.getMessage() + "n";
        strError += exp.getMessage() + "n";
        strError += "SQL: " + strSQL;
        System.out.println(strError);
    }
    return blnValid;
}
public FosterEligibility getFosterEligibility(String strUser) {

    String strSelectQuery = "SELECT WANT_FOSTER, IS_HEALTHY, 
    IS_HOME_SAFE, IS_FAMILY_INVOL, ATTEND_TRAINING 
    FROM foster_eligibility 
    WHERE ID = "+ strUser + ";"
    FosterEligibility fosterEligibility = new FosterEligibility();
    String strError = "";

    try {
        ResultSet rs = stmt.executeQuery(strSelectQuery);
        if(rs.next()) {
            fosterEligibility.setIsFosterRequired(rs.getString("WANT_FOSTER"));
            fosterEligibility.setIsHealth(rs.getString("IS_HEALTHY"));
            fosterEligibility.setIsHomeSafe(rs.getString("IS_HOME_SAFE"));
            fosterEligibility.setIsFamilyInvol(rs.getString("IS_FAMILY_INVOL"));
            fosterEligibility.setAttendTraining(rs.getString("ATTEND_TRAINING"));
        }
    } catch(Exception exp) {
        strError = "NJW-003: An error has occurred in getFosterEligibility() in WelfareDAO"
            + exp.getMessage() + ";"
        strError += exp.getMessage();
        strError += "\nQuery" + strSelectQuery;
        System.out.println(strError);
    }
    return fosterEligibility;
}

public WorkFirstBeneficiary getWorkFirstBeneficiary(String strUser) {

    String strSelectQuery = "SELECT * FROM workfirst_beneficiary 
    WHERE ID = " + strUser + ";"
    String strError = "";
    WorkFirstBeneficiary workFirstBeneficiary = new WorkFirstBeneficiary();

    WorkStatusDataBean workStatusDataBean = new WorkStatusDataBean();
    AbsentParentDataBean absentParentDataBean = new AbsentParentDataBean();
    TeenParentDataBean teenParentDataBean = new TeenParentDataBean();
    DrugProblemDataBean drugProblemDataBean = new DrugProblemDataBean();
try {
    ResultSet rs = stmt.executeQuery(strSelectQuery);
    if(rs.next()) {
        workFirstBeneficiary.setImmigrantStatus(rs.getString("immigrant_status"));
        workFirstBeneficiary.setDateEnterUSA(rs.getString("date_enter_usa"));
        workFirstBeneficiary.setNJRResidency(rs.getString("nj_residency"));
        workFirstBeneficiary.setHasChildren(rs.getString("has_children"));
        workFirstBeneficiary.setHasDrugProblem(rs.getString("has_drug_problem"));

        oracle.sql.STRUCT workStatus =
        (oracle.sql.STRUCT)rs.getObject("work_status");
        workStatusDataBean.setHasAssistance((String)object1Values[0]);
        workStatusDataBean.setAssistanceYear(String.valueOf(object1Values[1]));
        workStatusDataBean.setCurrentWorkStatus((String)object1Values[2]);
        workStatusDataBean.setWorkDeferReason((String)object1Values[3]);
        workFirstBeneficiary.setWorkStatusDataBean(workStatusDataBean);

        oracle.sql.STRUCT absentParent =
        (oracle.sql.STRUCT)rs.getObject("absent_parent");
        absentParentDataBean.setAbsentParentName((String)object2Values[0]);
        absentParentDataBean.setAbsentParentSSN((String)object2Values[1]);
        absentParentDataBean.setAbsentParentAddress((String)object2Values[2]);
        absentParentDataBean.setAbsentParentDOB((String)object2Values[3]);
        absentParentDataBean.setAbsentParentEmployer((String)object2Values[4]);
        absentParentDataBean.setAbsentParentVehicle((String)object2Values[5]);
        absentParentDataBean.setAbsentParentPlateNumber((String)object2Values[6]);
workFirstBeneficiary.setAbsentParentDataBean(absentParentDataBean);

Object[] object3Values = teenParent.getAttributes();

teenParentDataBean.setLiveAtHome((String)object3Values[0]);

teenParentDataBean.setFinishHighSchool((String)object3Values[1]);

teenParentDataBean.setAttendHighSchool((String)object3Values[2]);

workFirstBeneficiary.setTeenParentDataBean(teenParentDataBean);

Object[] object4Values = drugProblem.getAttributes();

drugProblemDataBean.setFelonyDrugDistribute((String)object4Values[0]);

drugProblemDataBean.setDrugPossesion((String)object4Values[1]);

drugProblemDataBean.setDrugTreatmentProgram((String)object4Values[2]);

workFirstBeneficiary.setDrugProblemDataBean(drugProblemDataBean);

} catch(Exception exp) {
    exp.printStackTrace();
    strError = "NJW-004: An error has occurred in getWorkFirstBeneficiary() in WelfareDAO"
        + exp.getMessage() + "\n";
    strError += exp.getMessage();
    strError += "\nQuery" + strSelectQuery;
    System.out.println(strError);
}

return workFirstBeneficiary;

}
try {
    ResultSet rs = stmt.executeQuery(strSelectQuery);
    if(rs.next()) {

        userBeneficiaryInfo.setNJResidency(rs.getString("nj_residency"));
        userBeneficiaryInfo.setHasInsurance(rs.getString("has_insurance"));

        oracle.sql.STRUCT beneficiary =
            (oracle.sql.STRUCT)rs.getObject("beneficiary");
        Object[] object1Values = beneficiary.getAttributes();
        beneficiaryDB.setSSN((String)object1Values[0]);
        beneficiaryDB.setName((String)object1Values[1]);
        beneficiaryDB.setSex((String)object1Values[2]);
        beneficiaryDB.setImmigrantStatus((String)object1Values[3]);
        beneficiaryDB.setEnterUsaDate((String)object1Values[4]);
        beneficiaryDB.setBirthDate((String)object1Values[5]);
        beneficiaryDB.setCountryOrigin((String)object1Values[6]);
        beneficiaryDB.setCountryCitigen((String)object1Values[7]);

        oracle.sql.STRUCT contact =
            (oracle.sql.STRUCT)object1Values[8];
        Object[] object11Values = contact.getAttributes();
        ContactType contactType = new ContactType();
        contactType.setEMail((String)object11Values[0]);
        contactType.setPhone(phone);
        contactType.setAddress(addressType);
        beneficiaryDB.setContact(contactType);

        userBeneficiaryInfo.setBeneficiaryDataBean(beneficiaryDB);
    }
}

userBeneficiaryInfo.setNJResidency(rs.getString("nj_residency"));
userBeneficiaryInfo.setHasInsurance(rs.getString("has_insurance"));

oracle.sql.STRUCT beneficiary =
    (oracle.sql.STRUCT)rs.getObject("beneficiary");
Object[] object1Values = beneficiary.getAttributes();
beneficiaryDB.setSSN((String)object1Values[0]);
beneficiaryDB.setName((String)object1Values[1]);
beneficiaryDB.setSex((String)object1Values[2]);
beneficiaryDB.setImmigrantStatus((String)object1Values[3]);
beneficiaryDB.setEnterUsaDate((String)object1Values[4]);
beneficiaryDB.setBirthDate((String)object1Values[5]);
beneficiaryDB.setCountryOrigin((String)object1Values[6]);
beneficiaryDB.setCountryCitigen((String)object1Values[7]);

oracle.sql.STRUCT contact =
    (oracle.sql.STRUCT)object1Values[8];
Object[] object11Values = contact.getAttributes();
ContactType contactType = new ContactType();
contactType.setEMail((String)object11Values[0]);
contactType.setPhone(phone);
contactType.setAddress(addressType);
beneficiaryDB.setContact(contactType);

userBeneficiaryInfo.setBeneficiaryDataBean(beneficiaryDB);
oracle.sql.STRUCT income =
(oracle.sql.STRUCT)rs.getObject("income");
Object[] object2Values = income.getAttributes();
try {
    System.out.println("---------------> " +
object2Values[0]);
    double dd =
Integer.parseInt(((String)object2Values[0]));
    System.out.println("dd --> " + dd);
    incomeEligibilityDB.setFamilySize(dd);
}catch(Exception e) {e.printStackTrace();}

incomeEligibilityDB.setFamilyIncome(Double.parseDouble((String)object2Values[1]));

userBeneficiaryInfo.setIncomeEligibilityDataBean(incomeEligibilityDB);

oracle.sql.STRUCT document =
(oracle.sql.STRUCT)rs.getObject("document");
Object[] object3Values = document.getAttributes();
NJDocTypeDataBean newNJDocTypeDataBean =
(docEligibilityTypeDB.setNJDocTypeDataBean((String)object3Values[0]));
IdentityDocDataBean newIdentityDocDataBean =
(docEligibilityTypeDB.setIdentityDocDataBean((String)object3Values[1]));
NutritionalDocDataBean newNutritionalDocDataBean =
(docEligibilityTypeDB.setNutritionalDocDataBean((String)object3Values[2]));

userBeneficiaryInfo.setDocEligibilityTypeDataBean(docEligibilityTypeDB);

benefitEligibilityDB.setBenefitEligibility(rs.getString("current_benefit"));
userBeneficiaryInfo.setBenefitEligibilityDataBean(benefitEligibilityDB);

} catch(Exception exp) {
    exp.printStackTrace();
    strError = "NJW-005: An error has occurred in getUserBeneficiaryInfo() in
WelFareDAO" + exp.getMessage() + "n",
    strError += exp.getMessage();
    strError += "nQuery" + strSelectQuery;
    System.out.println(strError);
    return userBeneficiaryInfo;
}

public boolean createWorkFirstBeneficiary(String strUser,
CIS 700B

string strImmigrantStatus,
string dateEnterUSA,
string strNJResidency,
WorkStatusDataBean workStatusDataBean,
string strHasChildren,
AbsentParentDataBean absentParentDataBean,
TeenParentDataBean teenParentDataBean,
string strHasDrugProblem,
DrugProblemDataBean drugProblemDataBean,
string strNewUser) {
    boolean blnValid = true;
    string strError = "";
    string strInsertQuery = "";
    try {
        if(strNewUser.equals("Old")) {
            strInsertQuery = "DELETE FROM workfirst_beneficiary
            WHERE ID = " + strUser + "";
            stmt.executeUpdate(strInsertQuery);
        }
        strInsertQuery = "INSERT INTO workfirst_beneficiary VALUES(" + strUser + ", "," + strImmigrantStatus + ", "," + dateEnterUSA + ", "," +
            work_status_type(" + workStatusDataBean.getHasAssistance() + ", "," +
            workStatusDataBean.getAssistanceYear() + ", "," +
            workStatusDataBean.getCurrentWorkStatus() + ", "," +
            workStatusDataBean.getWorkDeferReason() + ", "," +
            strHasChildren + ", "," +
            absent_parent_type(" + absentParentDataBean.getAbsentParentName() + ", "," +
            absentParentDataBean.getAbsentParentSSN() + ", "," +
            absentParentDataBean.getAbsentParentAddress() + ", "," +
            absentParentDataBean.getAbsentParentDOB() + ", "," +
            absentParentDataBean.getAbsentParentEmployer() + ", "," +
            absentParentDataBean.getAbsentParentVehicle() + ", "," +
            absentParentDataBean.getAbsentParentPlateNumber() + ", "," +
            teen_parent_type(" + teenParentDataBean.getLiveAtHome() + ", "," +
            teenParentDataBean.getFinishHighSchool() + ", "," +
            teenParentDataBean.getAttendHighSchool() + ", "," +
            "");
    }
    return blnValid;
}

public boolean createFosterEligibility(String strUser,
String strWantFoster,
String strIsHealthy,
String strIsHomeSafe,
String strIsFamilyInvol,
String strAttendTraining,
String strNewUser) {
    boolean blnValid = true;
    String strError = "";
    String strInsertQuery = "";
    try {
        if(strNewUser.equals("Old")) {
            strInsertQuery = "DELETE FROM foster_eligibility WHERE ID = " + strUser + ""
            stmt.executeUpdate(strInsertQuery);
        }
        strInsertQuery = "INSERT INTO foster_eligibility VALUES (" + strUser + ",
        + "'" + strWantFoster + ",
        + "'" + strIsHealthy + ",
        + "'" + strIsHomeSafe + ",
        + "'" + strIsFamilyInvol + ",
        + "'" + strAttendTraining + ")"
        int i = stmt.executeUpdate(strInsertQuery);
        if(i != 1)
            blnValid = false;
    } catch(Exception exp) {
        strError = "NJW-006: An error has occurred in
createWorkFirstBeneficiary() in WelFareDAO"
        + exp.getMessage() + "\n";
        strError += exp.getMessage();
        System.out.println(strError);
        System.out.println("Query: " + strInsertQuery);
        blnValid = false;
    }
    return blnValid;
}
```java
int i = stmt.executeUpdate(strInsertQuery);
if(i != 1)
    blnValid = false;
}
}

public boolean addUserBeneficiaryInfo(String strUser,
    BeneficiaryDataBean beneficiaryDB,
    IncomeEligibilityDataBean incomeEligibilityDB,
    BenifitEligibilityDataBean benifitEligibilityDB,
    DocEligibilityTypeDataBean docEligibilityTypeDB,
    String strNJResidency,
    String strHasInsurance,
    String strNewUser)
{
    boolean blnValid = true;
    String strError = "";
    String strInsertQuery = "";
    try{
        if(strNewUser.equals("Old")) {
            strInsertQuery = "DELETE FROM WIC_beneficiary
WHERE ID = '' + strUser + '''
            stmt.executeUpdate(strInsertQuery);
        }
        System.out.println("incomeEligibilityDB.getFamilySize() --> " + incomeEligibilityDB.getFamilySize());
        strInsertQuery = "INSERT INTO WIC_beneficiary VALUES(''
        + strUser + '''
        + "WIC_Beneficiary_type(''
        + "beneficiaryDB.getSSN() + "
        + "beneficiaryDB.getName() + "
        + "beneficiaryDB.getSex() + "
        + "beneficiaryDB.getImmigrantStatus() + "
        + "beneficiaryDB.getEnterUsaDate() + "
        + "beneficiaryDB.getBirthDate() + "
        + "beneficiaryDB.getCountryOrigin() + "
        + "beneficiaryDB.getCountryCitigen() + "
```

+ " contact_type(" +
beneficiaryDB.getEMail() + "," + "
+ " phones_type(" +
beneficiaryDB.getHome() + "," + beneficiaryDB.getWork() + "," +
beneficiaryDB.getCell() + "," + "
+ " address_type(" +
beneficiaryDB.getStreet() + "," + beneficiaryDB.getCity() + "," +
beneficiaryDB.getState() + "," + beneficiaryDB.getZIP() + "," +
beneficiaryDB.getCountry() + ")" + "
+ " " + benifitEligibilityDB.getBenifitEligibility() + "," + "
+ " doc_eligibility_type(" +
docEligibilityTypeDB.getNJDocTypeDataBean().getNJDocType() + "," +
docEligibilityTypeDB.getIdentityDocDataBean().getIdentityType() + "," +
docEligibilityTypeDB.getNutritionalDocDataBean().getNutritionalDoc() + ")" + "
+ " strNJResidency + "," + "
+ " strHasInsurance + 
"
int i = stmt.executeUpdate(strInsertQuery);
if(i != 1)
    blnValid = false;
} catch(Exception exp) {
    strError = "NJW-008: An error has occurred in addUserBeneficiaryInfo() in WelfareDAO"
    + exp.getMessage() + "\n";
    strError += exp.getMessage();
    System.out.println(strError);
    System.out.println("Query: " + strInsertQuery);
    blnValid = false;
}
return blnValid;
}

public boolean createFosterCarePreference(String strUser,
                                             String strGirl,
                                             String strBoy,
                                             String strEither,
                                             String strSibling,
                                             String strUpto5,
                                             String strUpto10,
strUpto12, String
strAbove13, String
strPhysical, String
strEmotional, String
strLearning, String
strExposure, String
strHIV, String
strBehavioural, String
strMedical, String
strDevelopment, String

strNewUser) {
    boolean blnValid = true;
    String strError = "";
    String strInsertQuery = "";
    try {
        if(strNewUser.equals("Old")) {
            strInsertQuery = "DELETE FROM care_preference_type
WHERE ID = " + strUser + ";
stmt.executeUpdate(strInsertQuery);
        }
        strInsertQuery = "INSERT INTO care_preference_type VALUES(" + strUser
+ "," + strGirl + "," + strBoy + "," + strEither + "," + strSibling + "," + strUpto5 + "," + strUpto10 + "," + strUpto12 + "," + strAbove13 + "," + strPhysical + "," + strEmotional + "," + strLearning + "," + strExposure + "," + strMedical + "," + strDevelopment + ")";
        int i = stmt.executeUpdate(strInsertQuery);
        if(i != 1)
            blnValid = false;
    } catch(Exception exp) {
        strError = "NJW-009: An error has occurred in
createFosterCarePreference() in WelFareDAO"
        + exp.getMessage() + "n";
    }
}
strError += exp.getMessage();
System.out.println(strError);
System.out.println("Query: " + strInsertQuery);
blnValid = false;
}
return blnValid;
}

public static void main(String s[]) {
    WelFareDAO welFareDAO = new WelFareDAO();
    System.out.println(welFareDAO.addUser("test", "test"));
    //welFareDAO.getFosterEligibility("ravi");
}

DataBean Files

AbsentParentDataBean.java

package databean;

public class AbsentParentDataBean {

    String strAbsentParentName        = "";
    String strAbsentParentSSN         = "";
    String strAbsentParentAddress     = "";
    String strAbsentParentDOB         = "";
    String strAbsentParentEmployer    = "";
    String strAbsentParentVehicle     = "";
    String strAbsentParentPlateNumber = "";

    public AbsentParentDataBean() {
    }

    public AbsentParentDataBean(String strAbsentParentName,
                                  String strAbsentParentSSN,
                                  String strAbsentParentAddress,
                                  String strAbsentParentDOB,
                                  String
                                  strAbsentParentEmployer,
                                  String strAbsentParentVehicle,
                                  String
                                  strAbsentParentPlateNumber) {
        this.strAbsentParentName        = strAbsentParentName;
        this.strAbsentParentSSN         = strAbsentParentSSN;
        this.strAbsentParentAddress     = strAbsentParentAddress;
        this.strAbsentParentDOB         = strAbsentParentDOB;
        this.strAbsentParentEmployer    = strAbsentParentEmployer;
        this.strAbsentParentVehicle     = strAbsentParentVehicle;
        this.strAbsentParentPlateNumber = strAbsentParentPlateNumber;
    }

Malathi Yellapragada
86
public void setAbsentParentName(String strAbsentParentName) {
    this.strAbsentParentName = strAbsentParentName;
}

public String getAbsentParentName() {
    return strAbsentParentName;
}

public void setAbsentParentSSN(String strAbsentParentSSN) {
    this.strAbsentParentSSN = strAbsentParentSSN;
}

public String getAbsentParentSSN() {
    return strAbsentParentSSN;
}

public void setAbsentParentAddress(String strAbsentParentAddress) {
    this.strAbsentParentAddress = strAbsentParentAddress;
}

public String getAbsentParentAddress() {
    return strAbsentParentAddress;
}

public void setAbsentParentDOB(String strAbsentParentDOB) {
    this.strAbsentParentDOB = strAbsentParentDOB;
}

public String getAbsentParentDOB() {
    return strAbsentParentDOB;
}

public void setAbsentParentEmployer(String strAbsentParentEmployer) {
    this.strAbsentParentEmployer = strAbsentParentEmployer;
}

public String getAbsentParentEmployer() {
    return strAbsentParentEmployer;
}

public void setAbsentParentVehicle(String strAbsentParentVehicle) {
    this.strAbsentParentVehicle = strAbsentParentVehicle;
}

public String getAbsentParentVehicle() {
    return strAbsentParentVehicle;
}

public void setAbsentParentPlateNumber(String strAbsentParentPlateNumber) {
    this.strAbsentParentPlateNumber = strAbsentParentPlateNumber;
}

public String getAbsentParentPlateNumber() {

return strAbsentParentPlateNumber;
}
}

**AddressType.java**

package databean;

public class AddressType {

    String strStreet = "";
    String strCity = "";
    String strState = "M";
    String strZIP = "";
    String strCountry = "";

    public AddressType() {
    }

    public AddressType(String strStreet,
                        String strCity,
                        String strState,
                        String strZIP,
                        String strCountry) {

        this.strStreet = strStreet;
        this.strCity = strCity;
        this.strState = strState;
        this.strZIP = strZIP;
        this.strCountry = strCountry;
    }

    public void setStreet(String strStreet) {
        this.strStreet = strStreet;
    }

    public String getStreet() {
        return strStreet;
    }

    public void setCity(String strCity) {
        this.strCity = strCity;
    }

    public String getCity() {
        return strCity;
    }

    public void setState(String strState) {

    }

}
BeneficiaryDataBean.java

package databean;

public class BeneficiaryDataBean {

    String strSSN = "";
    String strName = "";
    String strSex = "M";
    String strImmigrantStatus = "";
    String dateEnterUsa = null;
    String birthDate = null;
    String strCountryOrigin = "";
    String strCountryCitigen = "";
    ContactType contact = null;

    public BeneficiaryDataBean() {
    }

    public BeneficiaryDataBean(String strSSN, String strName, String strSex, String strImmigrantStatus, String dateEnterUsa, String birthDate, String strCountryOrigin, String strCountryCitigen, ContactType contact) {
    }

    public String strState = strState;
    
    public String getState() {
        return strState;
    }

    public void setZIP(String strZIP) {
        this.strZIP = strZIP;
    }

    public String getZIP() {
        return strZIP;
    }

    public void setCountry(String strCountry) {
        this.strCountry = strCountry;
    }

    public String getCountry() {
        return strCountry;
    }
}
String strCountryCitigen,
ContactType contact) {
this.strSSN = strSSN;
this.strName = strName;
this.strSex = strSex;
this.strImmigrantStatus = strImmigrantStatus;
this.dateEnterUsa = dateEnterUsa;
this.birthDate = birthDate;
this.strCountryOrigin = strCountryOrigin;
this.strCountryCitigen = strCountryCitigen;
this.contact = contact;
}

public void setSSN(String strSSN) {
    this.strSSN = strSSN;
}

public String getSSN() {
    return strSSN;
}

public void setName(String strName) {
    this.strName = strName;
}

public String getName() {
    return strName;
}

public void setSex(String strSex) {
    this.strSex = strSex;
}

public String getSex() {
    return strSex;
}

public void setImmigrantStatus(String strImmigrantStatus) {
    this.strImmigrantStatus = strImmigrantStatus;
}

public String getImmigrantStatus() {
    return strImmigrantStatus;
}

public void setBirthDate(String birthDate) {
    this.birthDate = birthDate;
}

public String getBirthDate() {
return birthDate;
}

public void setEnterUsaDate(String dateEnterUsa) {
    this.dateEnterUsa = dateEnterUsa;
}

public String getEnterUsaDate() {
    return dateEnterUsa;
}

public void setCountryOrigin(String strCountryOrigin) {
    this.strCountryOrigin = strCountryOrigin;
}

public String getCountryOrigin() {
    return strCountryOrigin;
}

public void setCountryCitigen(String strCountryCitigen) {
    this.strCountryCitigen = strCountryCitigen;
}

public String getCountryCitigen() {
    return strCountryCitigen;
}

public void setContact(ContactType contact) {
    this.contact = contact;
}

public ContactType getContact() {
    return contact;
}

//EMail ID
public void setEMail(String strEMail) {
    contact.strEMail = strEMail;
}

public String getEMail() {
    return contact.strEMail;
}

//Address Info
public void setStreet(String strStreet) {
    contact.objAddress.setStreet(strStreet);
}

public String getStreet() {
    return contact.objAddress.getStreet();
}
public void setCity(String strCity) {
    contact.objAddress.setCity(strCity);
}

public String getCity() {
    return contact.objAddress.getCity();
}

public void setState(String strState) {
    contact.objAddress.setState(strState);
}

public String getState() {
    return contact.objAddress.getState();
}

public void setZIP(String strZIP) {
    contact.objAddress.setZIP(strZIP);
}

public String getZIP() {
    return contact.objAddress.getZIP();
}

public void setCountry(String strCountry) {
    contact.objAddress.strCountry = strCountry;
}

public String getCountry() {
    return contact.objAddress.getCountry();
}

//Phone Info
public void setHome(String strHome) {
    contact.objPhone.setHome(strHome);
}

public String getHome() {
    return contact.objPhone.getHome();
}

public void setWork(String strWork) {
    contact.objPhone.setWork(strWork);
}

public String getWork() {
    return contact.objPhone.getWork();
}
public void setCell(String strCell) {
    contact.objPhone.setCell(strCell);
}

public String getCell() {
    return contact.objPhone.getCell();
}
}

class BenifitEligibilityDataBean.java

package databean;

public class BenifitEligibilityDataBean {
    String strBenifitEligibility = "N";

    /*
     * N = None
     * F = FoodStams
     * M = Medicaid
     * D = DependentChildren
     */

    public BenifitEligibilityDataBean() {
    }

    public BenifitEligibilityDataBean(String strBenifitEligibility) {
        this.strBenifitEligibility = strBenifitEligibility;
    }

    public void setBenifitEligibility(String strBenifitEligibility) {
        this.strBenifitEligibility = strBenifitEligibility;
    }

    public String getBenifitEligibility() {
        return strBenifitEligibility;
    }
}

CarePreferenceDataBean.java

package databean;

public class CarePreferenceDataBean {
    String strGirl = "";
    String strBoy = "";
    String strEither = "";
    String strSibling = "";

CIS 700B

String strUpto5 = "";
String strUpto10 = "";
String strUpto12 = "";
String strAbove13 = "";
String strPhysical = "";
String strEmotional = "";
String strLearning = "";
String strExposure = "";
String strHIV = "";
String strBehavioural = "";
String strMedical = "";
String strDevelopmental = "";

public CarePreferenceDataBean() {
}

public CarePreferenceDataBean(String strGirl,
    String strBoy,
    String strEither,
    String strSibling,
    String strUpto5,
    String strUpto10,
    String strUpto12,
    String strAbove13,
    String strPhysical,
    String strEmotional,
    String strLearning,
    String strExposure,
    String strHIV,
    String strBehavioural,
    String strMedical,
    String strDevelopment) {
    this.strGirl    = strGirl;
    this.strBoy    = strBoy;
    this.strEither    = strEither;
    this.strSibling    = strSibling;
    this.strUpto5    = strUpto5;
    this.strUpto10    = strUpto10;
    this.strUpto12    = strUpto12;
    this.strAbove13    = strAbove13;
    this.strPhysical    = strPhysical;
    this.strEmotional    = strEmotional;
    this.strLearning    = strLearning;
    this.strExposure    = strExposure;
    this.strHIV    = strHIV;
    this.strBehavioural    = strBehavioural;
    this.strMedical    = strMedical;
    this.strDevelopmental    = strDevelopmental;
}
public void setGirl(String strGirl) {
    this.strGirl = strGirl;
}

public String getGirl() {
    return strGirl;
}

public void setBoy(String strBoy) {
    this.strBoy = strBoy;
}

public String getBoy() {
    return strBoy;
}

public void setEither(String strEither) {
    this.strEither = strEither;
}

public String getEither() {
    return strEither;
}

public void setSibling(String strSibling) {
    this.strSibling = strSibling;
}

public String getSibling() {
    return strSibling;
}

public void setUpto5(String strUpto5) {
    this.strUpto5 = strUpto5;
}

public String getUpto5() {
    return strUpto5;
}

public void setUpto10(String strUpto10) {
    this.strUpto10 = strUpto10;
}

public String getUpto10() {
    return strUpto10;
}

public void setUpto12(String strUpto12) {
    this.strUpto12 = strUpto12;
}
public String getUpto12() {
    return strUpto12;
}

public void setAbove13(String strAbove13) {
    this.strAbove13 = strAbove13;
}

public String getAbove13() {
    return strAbove13;
}

public void setPhysical(String strPhysical) {
    this.strPhysical = strPhysical;
}

public String getPhysical() {
    return strPhysical;
}

public void setEmotional(String strEmotional) {
    this.strEmotional = strEmotional;
}

public String getEmotional() {
    return strEmotional;
}

public void setLearning(String strLearning) {
    this.strLearning = strLearning;
}

public String getLearning() {
    return strLearning;
}

public void setExposure(String strExposure) {
    this.strExposure = strExposure;
}

public String getExposure() {
    return strExposure;
}

public void setHIV(String strHIV) {
    this.strHIV = strHIV;
}

public String getHIV() {

return strHIV;
}

public void setBehavioural(String strBehavioural) {
    this.strBehavioural = strBehavioural;
}

public String getBehavioural() {
    return strBehavioural;
}

public void setMedical(String strMedical) {
    this.strMedical = strMedical;
}

public String getMedical() {
    return strMedical;
}

public void setDevelopment(String strDevelopmental) {
    this.strDevelopmental = strDevelopmental;
}

public String getDevelopment() {
    return strDevelopmental;
}

}

ContactType.java

package databean;

public class ContactType {

    String strEMail   = "";
    AddressType objAddress = null;
    PhoneType objPhone   = null;

    public ContactType() {
    }

    public ContactType(String strEMail,
                        AddressType objAddress,
                        PhoneType objPhone) {
        this.strEMail   = strEMail;
        this.objAddress = objAddress;
        this.objPhone   = objPhone;
    }

}
public void setEMail(String strEMail) {
    this.strEMail = strEMail;
}

public String getEMail() {
    return strEMail;
}

public void setAddress(AddressType objAddress) {
    this.objAddress = objAddress;
}

public AddressType getAddress() {
    return objAddress;
}

public void setPhone(PhoneType objPhone) {
    this.objPhone = objPhone;
}

public PhoneType getPhone() {
    return objPhone;
}

DocEligibilityTypeDataBean.java

package databean;

public class DocEligibilityTypeDataBean {

    NJDocTypeDataBean njDocTypeDataBean = null;
    IdentityDocDataBean identityDocDataBean = null;
    NutritionalDocDataBean nutritionalDocDataBean = null;

    public DocEligibilityTypeDataBean() {
    }

    public DocEligibilityTypeDataBean(NJDocTypeDataBean njDocTypeDataBean, IdentityDocDataBean identityDocDataBean, NutritionalDocDataBean nutritionalDocDataBean) {
        this.njDocTypeDataBean = njDocTypeDataBean;
    }
this.identityDocDataBean = identityDocDataBean;
this.nutritionalDocDataBean = nutritionalDocDataBean;
}

public void setNJDocTypeDataBean(NJDocTypeDataBean njDocTypeDataBean) {
    this.njDocTypeDataBean = njDocTypeDataBean;
}

public NJDocTypeDataBean getNJDocTypeDataBean() {
    return njDocTypeDataBean;
}

public void setIdentityDocDataBean(IdentityDocDataBean identityDocDataBean) {
    this.identityDocDataBean = identityDocDataBean;
}

public IdentityDocDataBean getIdentityDocDataBean() {
    return identityDocDataBean;
}

public void setNutritionalDocDataBean(NutritionalDocDataBean nutritionalDocDataBean) {
    this.nutritionalDocDataBean = nutritionalDocDataBean;
}

public NutritionalDocDataBean getNutritionalDocDataBean() {
    return nutritionalDocDataBean;
}

}
FosterEligibility.java

package databean;

public class FosterEligibility {

    String strIsHealth    = "Y";
    String strFosterRequired = "Y";
    String strIsHomeSafe  = "Y";
    String strIsFamilyInvol = "Y";
    String strAttendTraining = "Y";

    public FosterEligibility() {
    }

    public FosterEligibility(String strFosterRequired,
                            String strIsHealth,
                            String strIsHomeSafe,
                            String strIsFamilyInvol,
                            String strAttendTraining) {

    }

CIS 700B

this.strDrugTreatmentProgram = strDrugTreatmentProgram;
}

public void setFelonyDrugDistribute(String strFelonyDrugDistribute) {
    this.strFelonyDrugDistribute = strFelonyDrugDistribute;
}

public String getFelonyDrugDistribute() {
    return strFelonyDrugDistribute;
}

public void setDrugPossesion(String strDrugPossesion) {
    this.strDrugPossesion = strDrugPossesion;
}

public String getDrugPossesion() {
    return strDrugPossesion;
}

public void setDrugTreatmentProgram(String strDrugTreatmentProgram) {
    this.strDrugTreatmentProgram = strDrugTreatmentProgram;
}

public String getDrugTreatmentProgram() {
    return strDrugTreatmentProgram;
}
this.strFosterRequired = strFosterRequired;
this.strIsHealth = strIsHealth;
this.strIsHomeSafe = strIsHomeSafe;
this.strIsFamilyInvol = strIsFamilyInvol;
this.strAttendTraining = strAttendTraining;
}

public void setIsFosterRequired(String strFosterRequired) {
this.strFosterRequired = strFosterRequired;
}

public String getIsFosterRequired() {
return strFosterRequired;
}

public void setIsHealth(String strIsHealth) {
this.strIsHealth = strIsHealth;
}

public String getIsHealth() {
return strIsHealth;
}

public void setIsHomeSafe(String strIsHomeSafe) {
this.strIsHomeSafe = strIsHomeSafe;
}

public String getIsHomeSafe() {
return strIsHomeSafe;
}

public void setIsFamilyInvol(String strIsFamilyInvol) {
this.strIsFamilyInvol = strIsFamilyInvol;
}

public String getIsFamilyInvol() {
return strIsFamilyInvol;
}

public void setAttendTraining(String strAttendTraining) {
this.strAttendTraining = strAttendTraining;

public String getAttendTraining() {
return strAttendTraining;
}

IdentityDocDataBean.java
package databean;

public class IdentityDocDataBean {
    String strIdentityType   = "N";
    String strIdentityNumber = ""

    public IdentityDocDataBean() {
    }

    public IdentityDocDataBean(String strIdentityType) {
        this.strIdentityType   = strIdentityType;
    }

    public IdentityDocDataBean(String strIdentityType, String strIdentityNumber) {
        this.strIdentityType   = strIdentityType;
        this.strIdentityNumber = strIdentityNumber;
    }

    public void setIdentityType(String strIdentityType) {
        this.strIdentityType = strIdentityType;
    }

    public String getIdentityType() {
        return strIdentityType;
    }

    public void setIdentityNumber(String strIdentityNumber) {
        this.strIdentityNumber = strIdentityNumber;
    }

    public String getIdentityNumber() {
        return strIdentityNumber;
    }
}

IncomeEligibilityDataBean.java
package databean;

public class IncomeEligibilityDataBean {
    double familySize   = 0.0;
    double familyIncome = 0.0;

    public IncomeEligibilityDataBean() {
        
    }
public IncomeEligibilityDataBean(double familySize, double familyIncome) {
    this.familySize = familySize;
    this.familyIncome = familyIncome;
}

public void setFamilySize(double familySize) {
    this.familySize = familySize;
}

public double getFamilySize() {
    return familySize;
}

public void setFamilyIncome(double familyIncome) {
    this.familyIncome = familyIncome;
}

public double getFamilyIncome() {
    return familyIncome;
}

NJDocTypeDataBean.java

package databean;

public class NJDocTypeDataBean {
    String strNJDocType = "N";

    public NJDocTypeDataBean() {
    }

    public NJDocTypeDataBean(String strNJDocType) {
        this.strNJDocType = strNJDocType;
    }

    public void setNJDocType(String strNJDocType) {
        this.strNJDocType = strNJDocType;
    }

    public String getNJDocType() {

NutritionalDocDataBean.java

package databean;

public class NutritionalDocDataBean {

    String strNutritionalDoc = "N";

    public NutritionalDocDataBean() {
    }

    public NutritionalDocDataBean(String strNutritionalDoc) {
        this.strNutritionalDoc = strNutritionalDoc;
    }

    public void setNutritionalDoc(String strNutritionalDoc) {
        this.strNutritionalDoc = strNutritionalDoc;
    }

    public String getNutritionalDoc() {
        return strNutritionalDoc;
    }
}

PhoneType.java

package databean;

public class PhoneType {

    String strHome = "";
    String strWork = "";
    String strCell = "";

    public PhoneType() {
    }

    public PhoneType(String strHome, String strWork, String strCell) {
        this.strHome = strHome;
        this.strWork = strWork;
    }
}
this.strCell = strCell;
}

public void setHome(String strHome) {
    this.strHome = strHome;
}

public String getHome() {
    return strHome;
}

public void setWork(String strWork) {
    this.strWork = strWork;
}

public String getWork() {
    return strWork;
}

public void setCell(String strCell) {
    this.strCell = strCell;
}

public StringgetCell() {
    return strCell;
}

package databean;

public class TeenParentDataBean {

    String strLiveAtHome = ";
    String strFinishHighSchool = ";
    String strAttendHighSchool = ";

    public TeenParentDataBean() {
    }

    public TeenParentDataBean(String strLiveAtHome,
                                String strFinishHighSchool,
                                String strAttendHighSchool) {

        this.strLiveAtHome = strLiveAtHome;
        this.strFinishHighSchool = strFinishHighSchool;
    }
}
### UserBeneficiaryInfo.java

```java
package databean;

public class UserBeneficiaryInfo {

    BeneficiaryDataBean beneficiaryDB = new BeneficiaryDataBean();
    IncomeEligibilityDataBean incomeEligibilityDB = new IncomeEligibilityDataBean();
    BenifitEligibilityDataBean benifitEligibilityDB = new BenifitEligibilityDataBean();
    DocEligibilityTypeDataBean docEligibilityTypeDB = new DocEligibilityTypeDataBean();

    String strNJResidency = "";
    String strHasInsurance = "";

    public UserBeneficiaryInfo() {
    
    }

    public UserBeneficiaryInfo(BeneficiaryDataBean beneficiaryDB,
        IncomeEligibilityDataBean incomeEligibilityDB,
        BenifitEligibilityDataBean benifitEligibilityDB,
        DocEligibilityTypeDataBean docEligibilityTypeDB,
        String strNJResidency, String strHasInsurance) {
    
    }

}
```
DocEligibilityTypeDataBean docEligibilityTypeDB,
String strNJResidency,
String strHasInsurance) {

beneficiaryDB;
incomeEligibilityDB;
benefitEligibilityDB;
docEligibilityTypeDB;
strNJResidency;
strHasInsurance;

}
public void setBenefitEligibilityDataBean(BenefitEligibilityDataBean benifitEligibilityDB) {
    this.benifitEligibilityDB = benifitEligibilityDB;
}

public BenefitEligibilityDataBean getBenefitEligibilityDataBean() {
    return benifitEligibilityDB;
}

public void setDocEligibilityTypeDataBean(DocEligibilityTypeDataBean docEligibilityTypeDB) {
    this.docEligibilityTypeDB = docEligibilityTypeDB;
}

public DocEligibilityTypeDataBean getDocEligibilityTypeDataBean() {
    return docEligibilityTypeDB;
}

WorkFirstBeneficiary.java

package databean;

public class WorkFirstBeneficiary {

    String strImmigrantStatus = "";
    String strDateEnterUSA = "";
    String strNJResidency = "";
    String strHasChildren = "";
    String strHasDrugProblem = "";
    WorkStatusDataBean workStatusDataBean = new WorkStatusDataBean();
    AbsentParentDataBean absentParentDataBean = new AbsentParentDataBean();
    TeenParentDataBean teenParentDataBean = new TeenParentDataBean();
    DrugProblemDataBean drugProblemDataBean = new DrugProblemDataBean();

    public WorkFirstBeneficiary() {
    }

    public WorkFirstBeneficiary(String strUser,
            String strImmigrantStatus,
            String strDateEnterUSA,
            String strNJResidency,
            String strHasChildren,
            String strHasDrugProblem) {
        this.strImmigrantStatus = strImmigrantStatus;
        this.strDateEnterUSA = strDateEnterUSA;
        this.strNJResidency = strNJResidency;
        this.strHasChildren = strHasChildren;
        this.strHasDrugProblem = strHasDrugProblem;
        workStatusDataBean = new WorkStatusDataBean();
        absentParentDataBean = new AbsentParentDataBean();
        teenParentDataBean = new TeenParentDataBean();
        drugProblemDataBean = new DrugProblemDataBean();
    }

String strHasDrugProblem,
WorkStatusDataBean workStatusDataBean,
AbsentParentDataBean absentParentDataBean,
TeenParentDataBean teenParentDataBean,
DrugProblemDataBean drugProblemDataBean) {

    this.strImmigrantStatus = strImmigrantStatus;
    this.strDateEnterUSA = strDateEnterUSA;
    this.strNJResidency = strNJResidency;
    this.strHasChildren = strHasChildren;
    this.strHasDrugProblem = strHasDrugProblem;
    this.workStatusDataBean = workStatusDataBean;
    this.absentParentDataBean = absentParentDataBean;
    this.teenParentDataBean = teenParentDataBean;
    this.drugProblemDataBean = drugProblemDataBean;
}

public void setImmigrantStatus(String strImmigrantStatus) {
    this.strImmigrantStatus = strImmigrantStatus;
}

public String getImmigrantStatus() {
    return strImmigrantStatus;
}

public void setDateEnterUSA(String strDateEnterUSA) {
    this.strDateEnterUSA = strDateEnterUSA;
}

public String getDateEnterUSA() {
    return strDateEnterUSA;
}

public void setNJResidency(String strNJResidency) {
    this.strNJResidency = strNJResidency;
}

public String getNJResidency() {
    return strNJResidency;
}

public void setHasChildren(String strHasChildren) {
    this.strHasChildren = strHasChildren;
}

public String getHasChildren() {
    return strHasChildren;
}
public class WorkStatusDataBean {
    String strHasAssistance = "Y";
}

WorkStatusDataBean.java

databean;

package databean;

class WorkStatusDataBean {
    String strHasAssistance = "Y";
}
String strAssistanceYear = "";
String strCurrentWorkStatus = "";
String strWorkDeferReason = "";

public WorkStatusDataBean() {
}

public WorkStatusDataBean(String strHasAssistance,
String strAssistanceYear,
String strCurrentWorkStatus,
String strWorkDeferReason)
{
    this.strHasAssistance = strHasAssistance;
    this.strAssistanceYear = strAssistanceYear;
    this.strCurrentWorkStatus = strCurrentWorkStatus;
    this.strWorkDeferReason = strWorkDeferReason;
}

public void setHasAssistance(String strHasAssistance) {
    this.strHasAssistance = strHasAssistance;
}

public String getHasAssistance() {
    return strHasAssistance;
}

public void setAssistanceYear(String strAssistanceYear) {
    this.strAssistanceYear = strAssistanceYear;
}

public String getAssistanceYear() {
    return strAssistanceYear;
}

public void setCurrentWorkStatus(String strCurrentWorkStatus) {
    this.strCurrentWorkStatus = strCurrentWorkStatus;
}

public String getCurrentWorkStatus() {
    return strCurrentWorkStatus;
}
public void setWorkDeferReason(String strWorkDeferReason) {
    this.strWorkDeferReason = strWorkDeferReason;
}

public String getWorkDeferReason() {
    return strWorkDeferReason;
}
}