**Software Design**

**Document**

**for**

**Referral / ERRA Trending Analysis Tool (R/E TAT)**

**Version 2.0 approved**

**Prepared by Wilson Thomas**

**Emanoel K.**

**Henry Liwag**

**Daniel Limas**

**Xiaoye Li**

**Philip Tran**

**QTC Management / Grace Pang-Truong**

**Preliminary version: April 14, 2018**

Table of Contents................................................................................................<pg 1-2>

Revision History............................................................................................<pg 3>

Introduction.........................................................................................<pg 4>

* 1. Purpose......................................................................................<pg 4>
  2. Document Conventions…………………….........................<pg 4>
  3. Intended Audience and Reading Suggestions...........................<pg 4>
  4. System Overview..........................................................................<pg 4>

1. Design Considerations.....................................................................................<pg 5>
   1. Assumptions and dependencies......................................................<pg 5>
   2. General Constraints..................................................................<pg 5>
   3. Goals and Guidelines..............................................................<pg 5>
   4. Development Methods........................................................<pg 6>
2. 3. Architectural Strategies..........................................................................<pg 7-8>
3. 4. System Architecture...................................................................................<pg 9>
   1. ....................................................................................<pg 10>
   2. ...................................................................................................<pg 10-11>
4. 6. Detailed System Design........................................................................<pg 11>

6.1 User Interface Module..................................................................<pg 11>

6.1.1 Responsibilities..................................................................<pg 11>

6.1.2 Constraints......................................................................<pg 11>

6.1.3 Composition.............................................................<pg 11>

6.1.4 Uses/Interactions..............................................................<pg 11>

6.1.5 Resources................................................................<pg 11>

6.2 Main Controller Module

6.2.1 Responsibilities..................................................................<pg 12>

6.2.2 Constraints......................................................................<pg 12>

6.2.3 Composition.............................................................<pg 12>

6.2.4 Uses/Interactions..............................................................<pg 12>

6.2.5 Resources................................................................<pg 12>

6.3 Data Input Parser Module

6.3.1 Responsibilities..................................................................<pg 12>

6.3.2 Constraints......................................................................<pg 12>

6.3.3 Composition.............................................................<pg 13>

6.3.4 Uses/Interactions..............................................................<pg 13>

6.3.5 Resources................................................................<pg 14>

6.4 Authentication

6.4.1 Responsibilities..................................................................<pg 14>

6.4.2 Constraints......................................................................<pg 14>

6.4.3 Composition.............................................................<pg 14>

6.4.4 Uses/Interactions..............................................................<pg 14>

6.4.5 Resources................................................................<pg 14>

1. 8. User Interface

8.1 Overview of User Interface........................................................<pg 15>

8.2 Screen Frameworks or Images..............................................<pg 15-19>

8.3 User Interface Flow Model...........................................................<pg 20>

1. 11. Glossary...................................................................................................<pg 21>
2. 12. References............................................................................................<pg 21>

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Wilson Thomas | 12/8/17 | Prelimanary version | 1.0 |
| Wilson Thomas | 4/12/18 | Updates | 2.0 |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1 Purpose**

This document will outline in detail the Referral / ERRA Trending Analysis Tool (RE- TAT)’s software architecture and design. This document will display the system’s design from several viewpoint to provide a guide into how the system works and to communicate what the system does. It intends to get an insight into the architectural and design decisions that were made for the RE-TAT.

**1.2 Document Conventions**

This document follows APA-like format, with bold-faced text used to emphasize the section headings and 12-sized Times New Roman font. Bullet points have been used to quickly identify sub-sections and organize the document to be read easily. This document attempts to outline the requirements of the application clearly and accurately. It also outlines the expected behavior of the application.

**1.3 Intended Audience and Reading Suggestions**

This document is written on a technical level to address the QTC Management team, QTC IT team, and Cal State LA computer science department.

**1.4 System Overview**

This document provides the architecture and design of the RE-TAT. Given several sets of data, RE-TAT will create a trending analysis tool that will identify areas what type of specialty and how many claimants are in the area, validate appointment assigned to QTC, and determine if an area has enough providers within a given distance. This software will allow the QTC management team to effectively balance how the workload will be allocated across the provider areas.

**2. Design Considerations**

**2.1 Assumptions and Dependencies**

* Possible changes in functionality after delivery date
* Possible changes in the way users input data
* Software depends on external files by user
* Software depends on user input
* Software depends on an internal system files list
* Software depends on external files formatting

**2.2 General Constraints**

* 5GB caches in Apache Tomcat for file reading
* Zipcode excel file will not be up-to-date with future changes to US zipcodes
* There are only four QTC Excel file formats the RE-TAT system is responsible to read.
* Software can only be view through an internet connection

**2.3 Goals and Guidelines**

* Let users know that claimants assign to QTC matches with the referral data.
* Let users know that claimants not assign to QTC are within the referral data
* Let users know that claimants that are suppose to be assign to QTC are not within the referral data
* The software should be able to list the specialties that are and are not covered around several miles around a given location
* The software should be able to give user access to selecting specialty to search upon.
* The software should be able to give user access to modify the specialty mileage range.
* This software should be designed with future teams in mind.
  + Future teams should easily understanding the source code and data flow of the software
  + Allow future teams to easily add new features

**2.4 Development Methods**

This project initially used the Waterfall Development method in order to measure the progress of the deliverable. Due to the amount of data and constant changes to the requirement and functionally, we switched to agile development method.

**3. Architectural Strategies**

* In the development of RE-TAT, development tools will include the following:
  + Java Oracle
  + Eclipse IDE
  + Tomcat Apache 7
  + MySQL
  + phpMyAdmin
  + Bootstrap
  + Apache POI Library
* RETAT will be using the MVC (Model-View-Controller) architecture in order to simplify system implementation. The controller module of MVC will be the central dispatcher for different modules throughout the system.
* CSS libraries, such as Bootstrap, etc. will create the user interface, as well as to modularize each section of the JSP files.
* Admin user has the ability to create user and login to access the website and the databases.
* Input verification technologies implemented using form validation techniques will be used in order to verify the integrity of the dataset being inputted by the user. This is important especially during the cases when the user must input data from a form. Users could inadvertently execute commands through forms without some way of cleaning or deleting the input.
* One administrator or members of the team with appropriate access privileges will be able to make updates on the system's internal Excel files. Regular users who will log in to the server are to be authenticated on the log-in system and be registered as a normal user.
* For concurrency and synchronization, multiple users are allowed to access the website. RE-TAT would be able to support as much sessions as the server allows.
* The web application will use the HTTPS protocol to ensure that sensitive data remains secure when the output is being sent to the user.

**4. System Architecture**

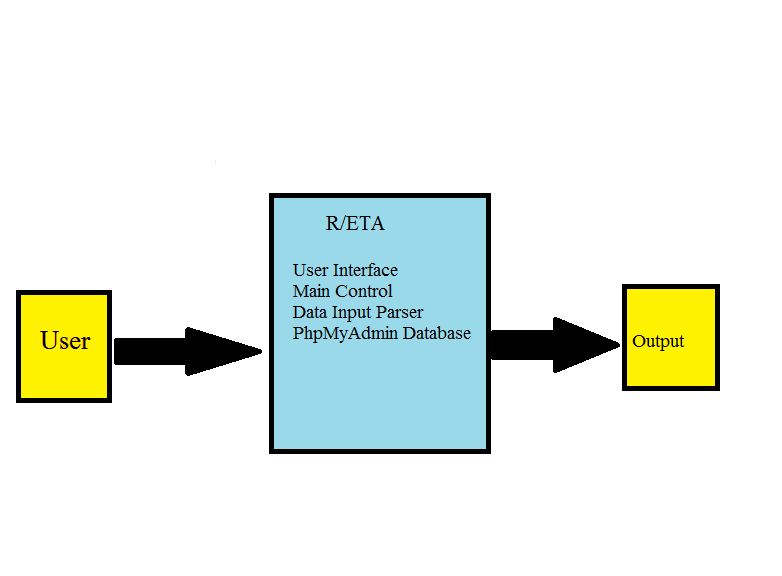


Figure 4.0

In Figure 4.0, displays the data flow diagram of level 0. Before using the tool the user must interact with the tool. The arrow indicates input to the system of modules. Each element of the list is a module. When user enters an input, the modules interact with the input data.

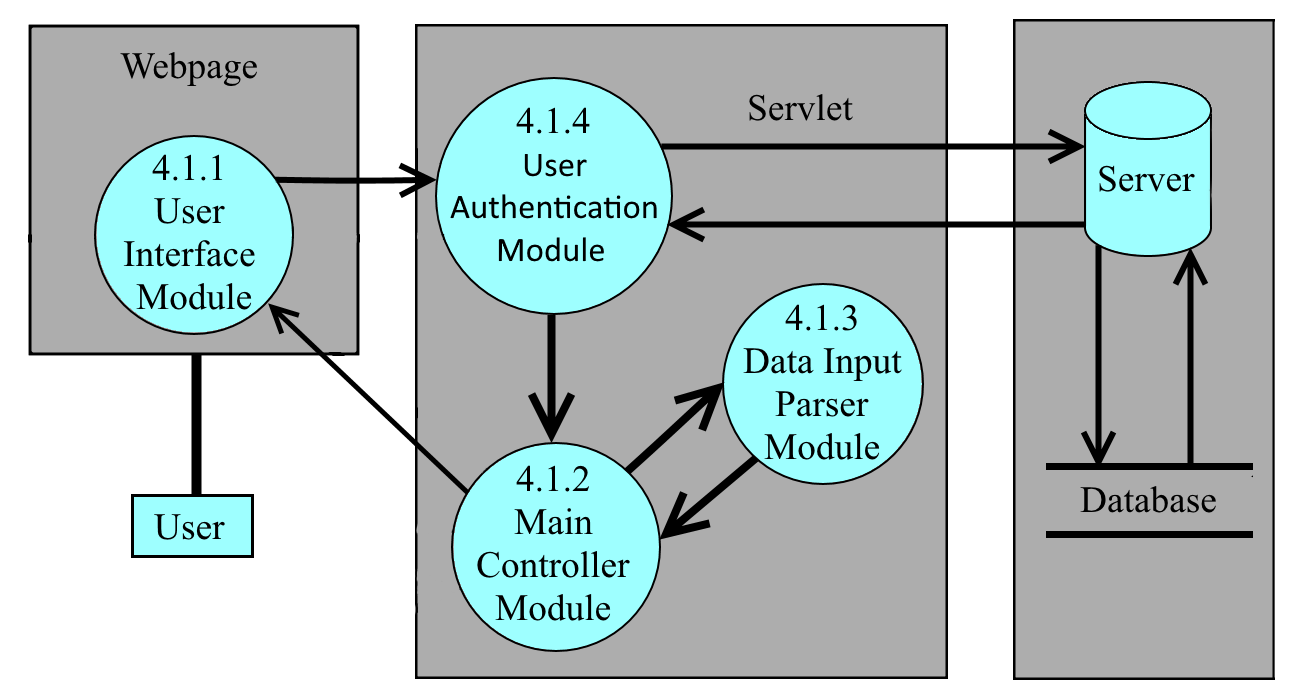


Figure 4.1

**4.1.1 User Interface Module**

The User Interface Module(UIM) provides a web-base GUI and functionality for the user-friendly interface. It organizes and visualizes the data-set based on user geolocation input, defined categories’ input, and distance input. The UIM contains tools such as searching geo-location based on user location input and time of date and pinpointing geo-location. The UIM also contains feature that identifies specialty availability within a region and determine how many claimants are in the area, validate appointment assigned to QTC, and determine if an area has enough providers within a given distance. For security, the user will to enter their login information on the UIM to gain access to the system. The UIM is also responsible for receiving the QTC’s excel file information and sending it to the UIM.

**4.1.2 Main Controller Module**

The Main Controller Module(MCM) is the Model View Controller(MVC) of the RE-TAT. M stands for models. Which when the data is parsed it creates model’s so the main controller module can manipulate the data. V stands for View, and the User Interface Module is that part of the system’s design pattern. View is will in charge of display and having a way to send input to the Main Controller Module. C stands for controller and the Main Controller Module is the controller for the view. It contains the central system of the server side implementation on the RE-TAT. The MVC will serve as the main controller for the User Interface Module(UIM), Data Input Parser Module(DIPM), User Module(UM) and the server. The MCM is responsible for the implementation of the features. Based on the user’s input of date, geolocation, medical specialty for a given feature from the UIM, the MCM will search through the dataset for a list of information the user wants to view and sends it to the UIM. The MCM is also responsible for transferring Excel files to the DIPM and receiving the content of the Excel files' data from the DIPM.

**4.1.3 Data Input Parser Module.**

The Data Input Parser Module receives input files and parses the data from the Main Controller Module and creates data models for the parsed information. After the files have been parsed, the module will that data to the Main Controller module. The higher-level components work together as whole and follow the MVC design pattern.

**4.1.3 User Authentication Module**

User Authentication Module connects to a database which store all username and password information. General users will use their unique username and password to access this program, but they will not have access to admin level privilege. There is only one Admin user in our system. Admin user have the same abilities as the general user as well as add and remove users, and change the users’ password.

**6. Detailed System Design (DFD Level 2)**

**6.1 User Interface Module**

**6.1.1 Responsibilities**

Referral and Provider check feature allows QTC to check whether there are areas that are missing providers. The filter can be used to filter based on state and also find specific Zip Code. There is also an option to filter whether the resulting Zip Code meets the specialty requirement. Erra and Provider cross check feature helps QTC check between their referral list and ERRA data to see what part of the data matches and what doesn’t match.

**6.1.2 Constraints**

Since user can type input into the textbox, the system has sanitize their input.

**6.1.3 Composition**

Filters in our features is a meaningful subcomponents that helps QTC narrow down their results from their data.

**6.1.4 Uses/Interactions**

The user will use our interactive website by using the hierarchy options we implemented that manipulates the data we were given by QTC.

**6.1.5 Resources**

The main resource we have is QTC Excel files. Due to security and privacy concerns, our data is limited but it gives us the necessary data needed to execute the desired goals of helping QTC identify areas where they are missing providers and help them know when and where are referrals were assigned.

**6.2 Main Controller Module**

**6.2.1 Responsibilities**

This component is one of the cores of the system. The module handles three sub modules depending on the user’s interaction. It sends useful information to the client. This module is the controller of the User Interface Module. This module will control the input given by the user and send output to the user.

**6.2.2 Constraints**

An assumption is that passing a large Excel file will take some time for the module to send back the data.

**6.2.3 Composition**

The module will help the client know how many claimants are in the area, validate appointment assigned to QTC, and determine if an area has enough providers within a given distance.

**6.2.4 Uses/Interactions**

The Main Controller Module receives input from the User Interface Module and sends it back to the User Interface Module. If the MCM receives an input file from the UAM then MCM will send the file to the Data Input Parser Module. Then, MCM receives output from DIPM of the data model. Inside MCM there are four submodules. Each module receives input depending on the UIM. Only three submodules will receive output from the DIPM.

**6.2.5 Resources**

* JDBC library

**6.3 Data Input Parser Module**

**6.3.1 Responsibilities**

The Data Input Parser Module reads in QTC’s Excel files and outputs data models the RE-TAT system can understand and operate on. This module serves as the main interpreter between the QTC’s Excel data format and the RE-TAT’s data model format.

**6.3.2 Constraints**

* QTC’s excel file formats are a constraint on the Data Input Parser Module. Currently, there are four formats this system can read, parser and understand. If QTC changes the formatting of their Excel files, the module would not be able to interpret it and may cause output issue to the UIM or crash the webpage.
* The Data Input Parser Module might take about 5-10 second to read and parse the Excel files. Some of the files may contain more than 200,000 data entries. Issues about memory storage and wait times can arise when the number of data entries increases.

**6.3.3 Composition**

There are four subcomponents in the Data Input Parser Module: QTC’s ERRA excel file formatter, QTC’s referral list excel file formatter, QTC’s VA provider network list Excel file formatter and QTC’s specialty and mileage range excel file formatter.

* ERRA Excel file formatter
  + This subcomponents translate the QTC’s ERRA excel file data into a dataset the RE-TAT system can interpret.
* Referral list excel file formatter
  + This subcomponents translate the QTC’s referral list excel file data into a dataset the RE-TAT system can interpret.
* VA provider network list excel file formatter
  + This subcomponents translate the QTC’s VA provider network list excel file data into a dataset the RE-TAT system can interpret.
* Specialty and mileage range excel file formatter
  + This subcomponents translate the QTC’s specialty and mileage range excel file data into a dataset the RE-TAT system can interpret.

**6.3.4 Uses/Interactions**

* The Data Input Parser Module only communicate with the Main Controller Module in the RE-TAT. The DIPM received the Excel files from the MCM and sent the data model back to the MCM.
* There is a wait time between when the user sends the excel file and when the RE-TAT’s UI displays the results. The wait time comes from the reading and parsing of the Excel files in the DIPM.

**6.3.5 Resources**

Apache POI is an external java library resource used in this software to read the Excel files. Depending on the size of the Excel file, Apache POI might use as much or little of the machine system’s memory.

**6.4 User Authentication Module**

**6.4.1 Responsibilities**

User Authentication Module control the access of this program. Different level of user have different access of this program.

**6.4.2 Constraints**

User Authentication Module must have internet connection to verify user’s authentication.

**6.4.3 Composition**

Admin submodule allow the admin user to add and delete users and change user’s password.

**6.4.4 Uses/Interactions**

After the User Authentication Module verified user’s identity, it will direct users to the Main Controller Module

**6.4.5 Resources**

User Authentication Module has its own database to store user information**.** To connect to this database, User Authentication Module uses JDBC connector library.

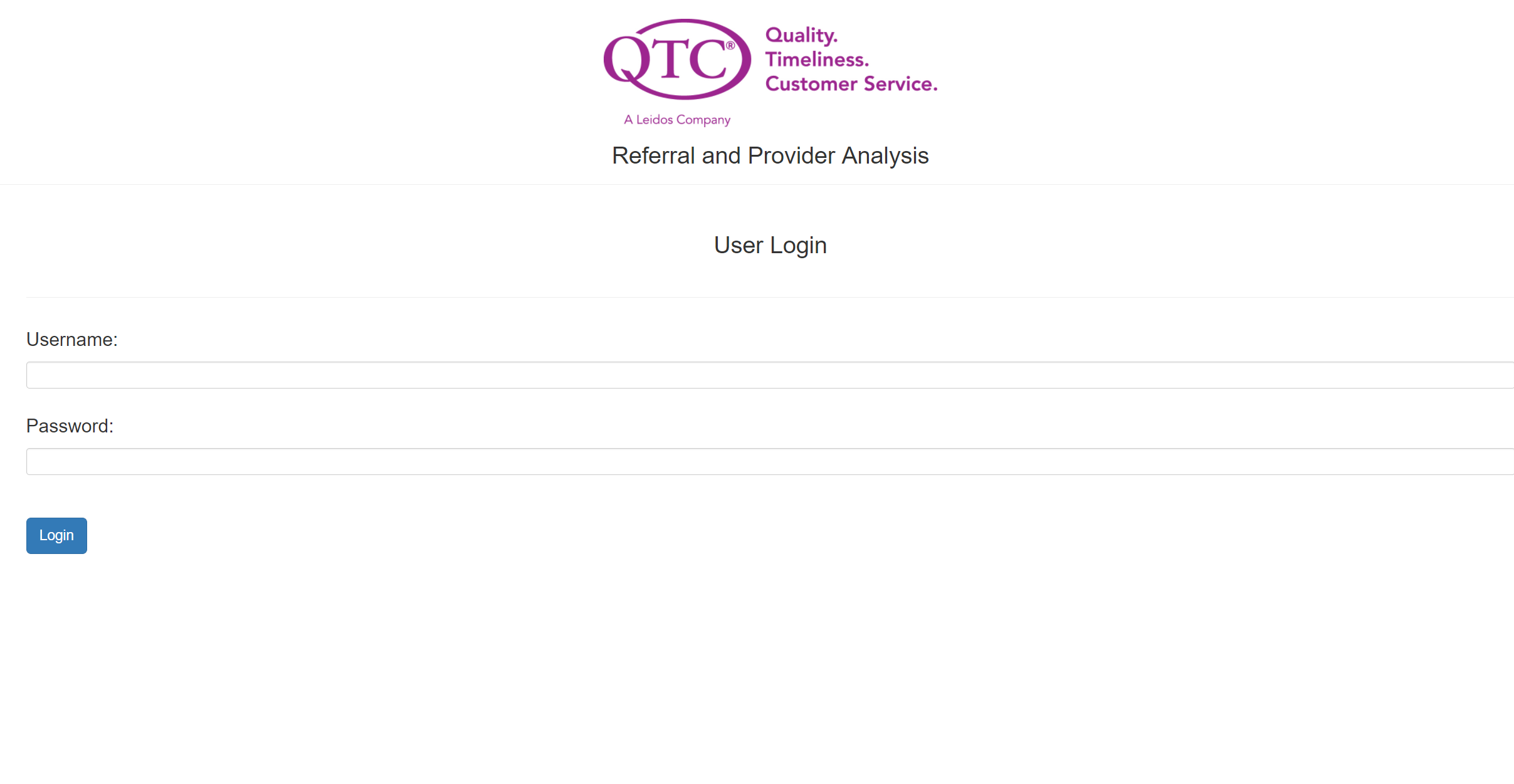
**8. User Interface**

**8.1 Overview of User Interface**

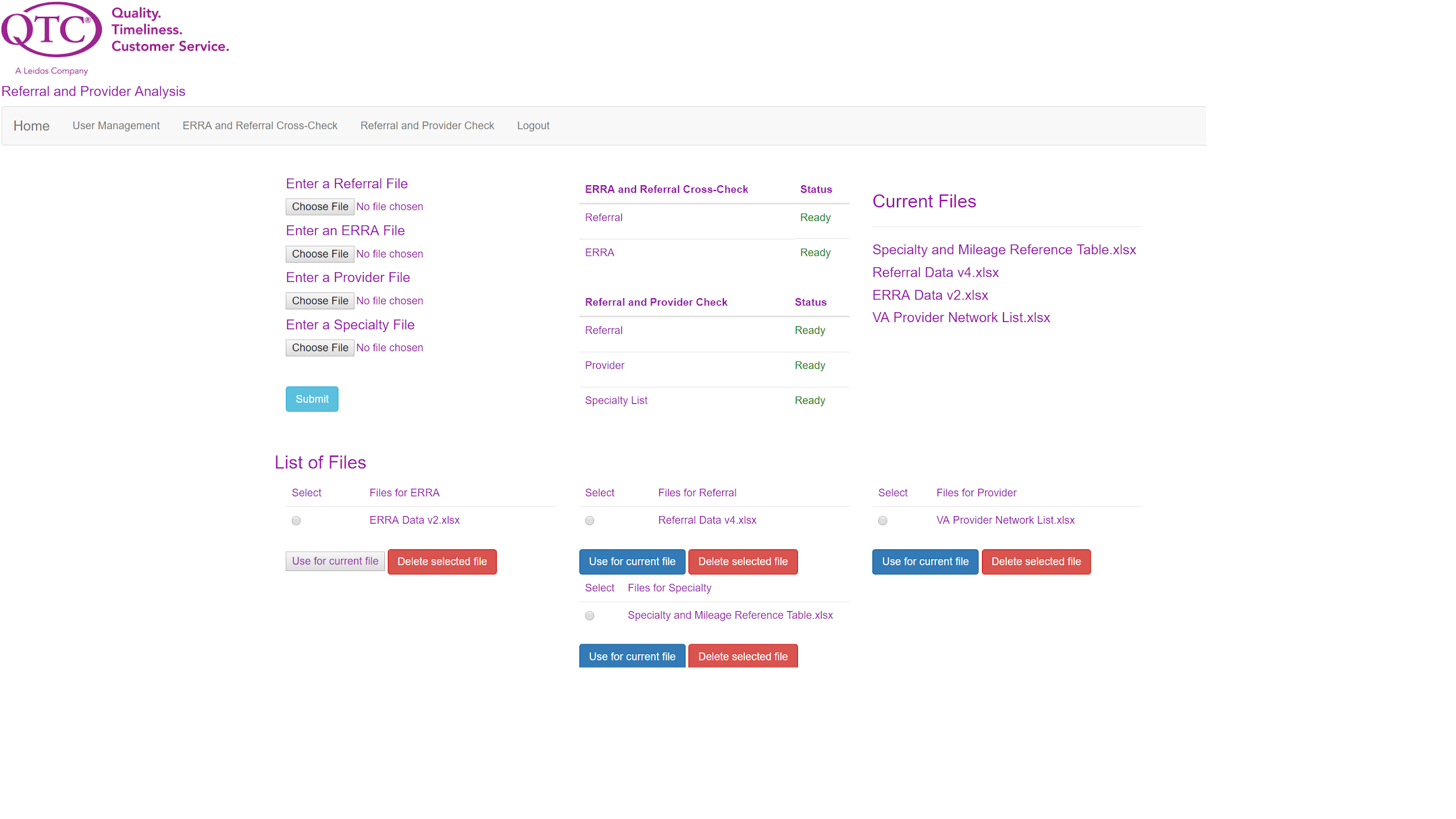
Upon visiting the site the user will be at the homepage with a list of the features. The user then will be able to navigate to each specific feature page from the home page. Each feature page will have a filter which will allow the user to broaden or narrow the result down based on the user's need. The user will also be able to search for specific results they are looking for. For each of the results the user will then have the option to visit the result specific page will show more information on this result.

**8.2 Screen Frameworks or Images**

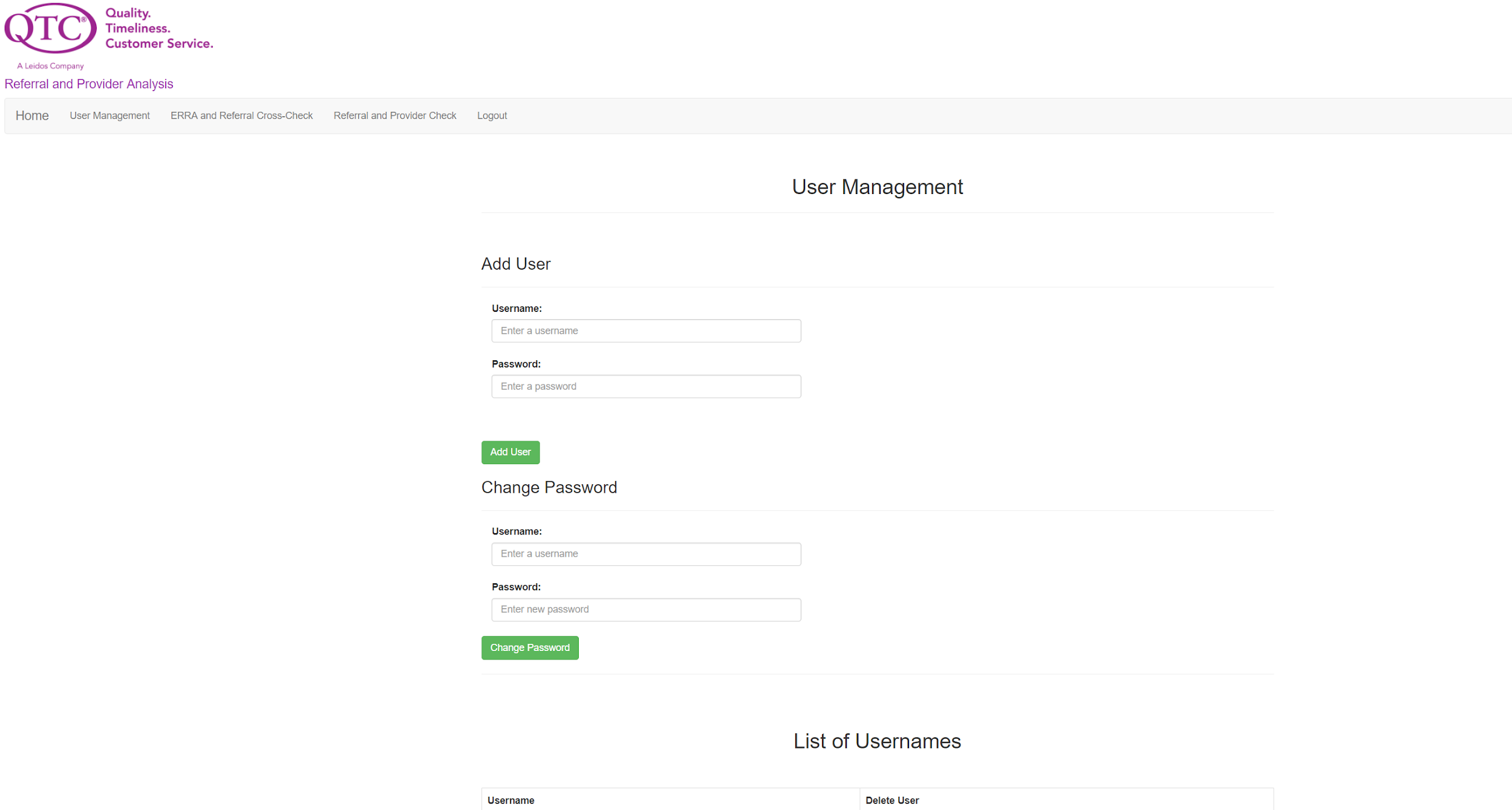
Actual screenshots of the various UI screens and popups.



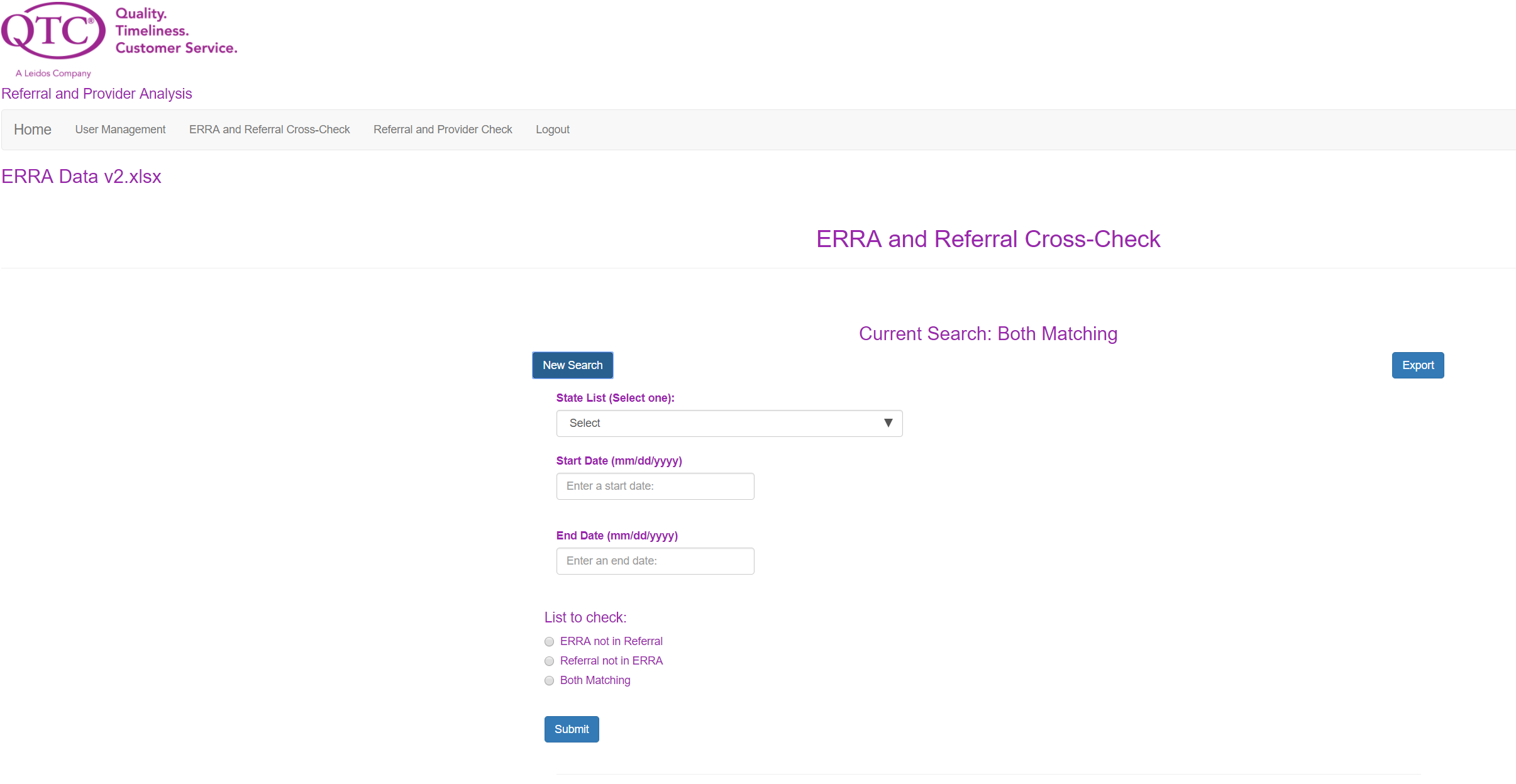
**Home Page**

****

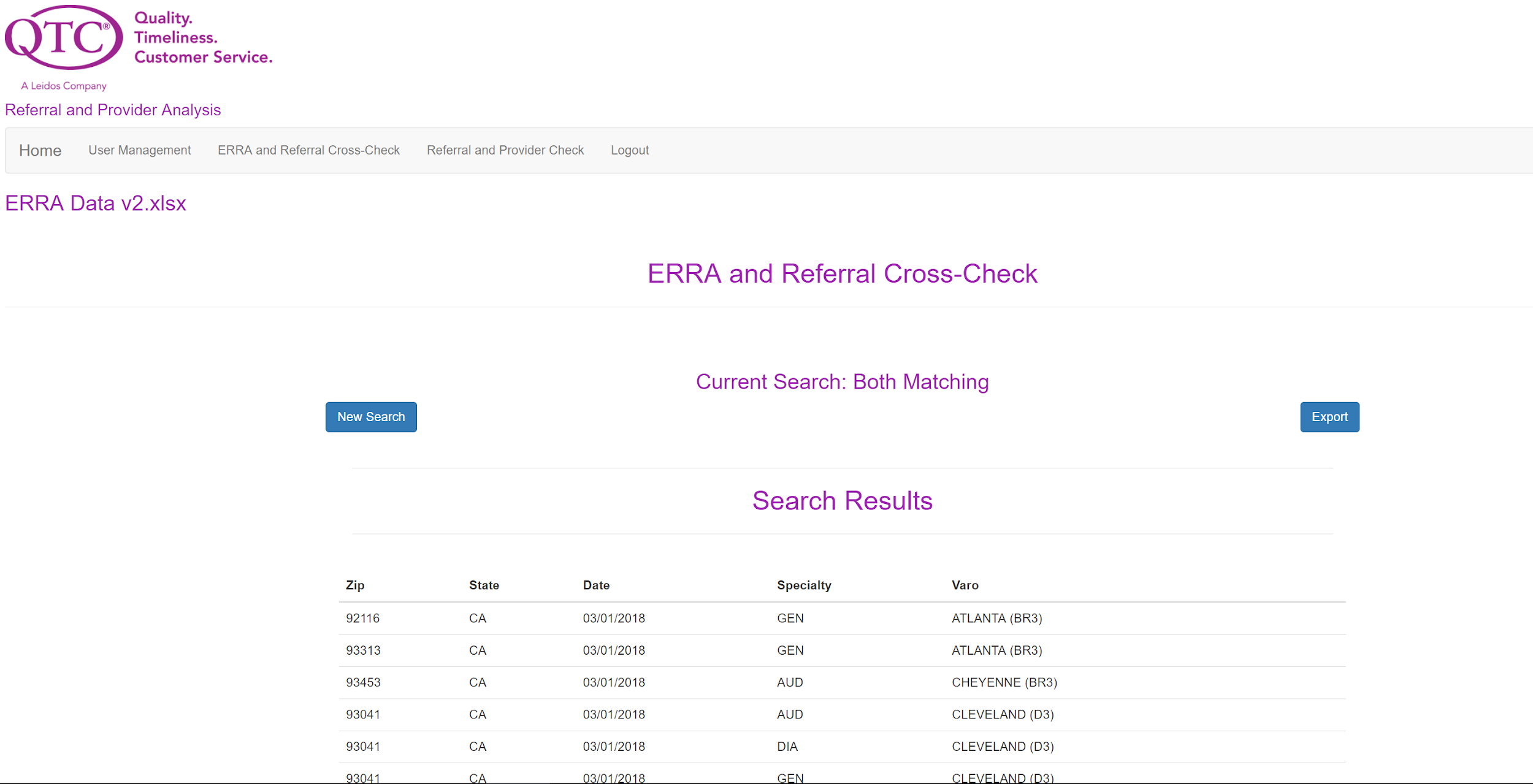
**User Management**

****

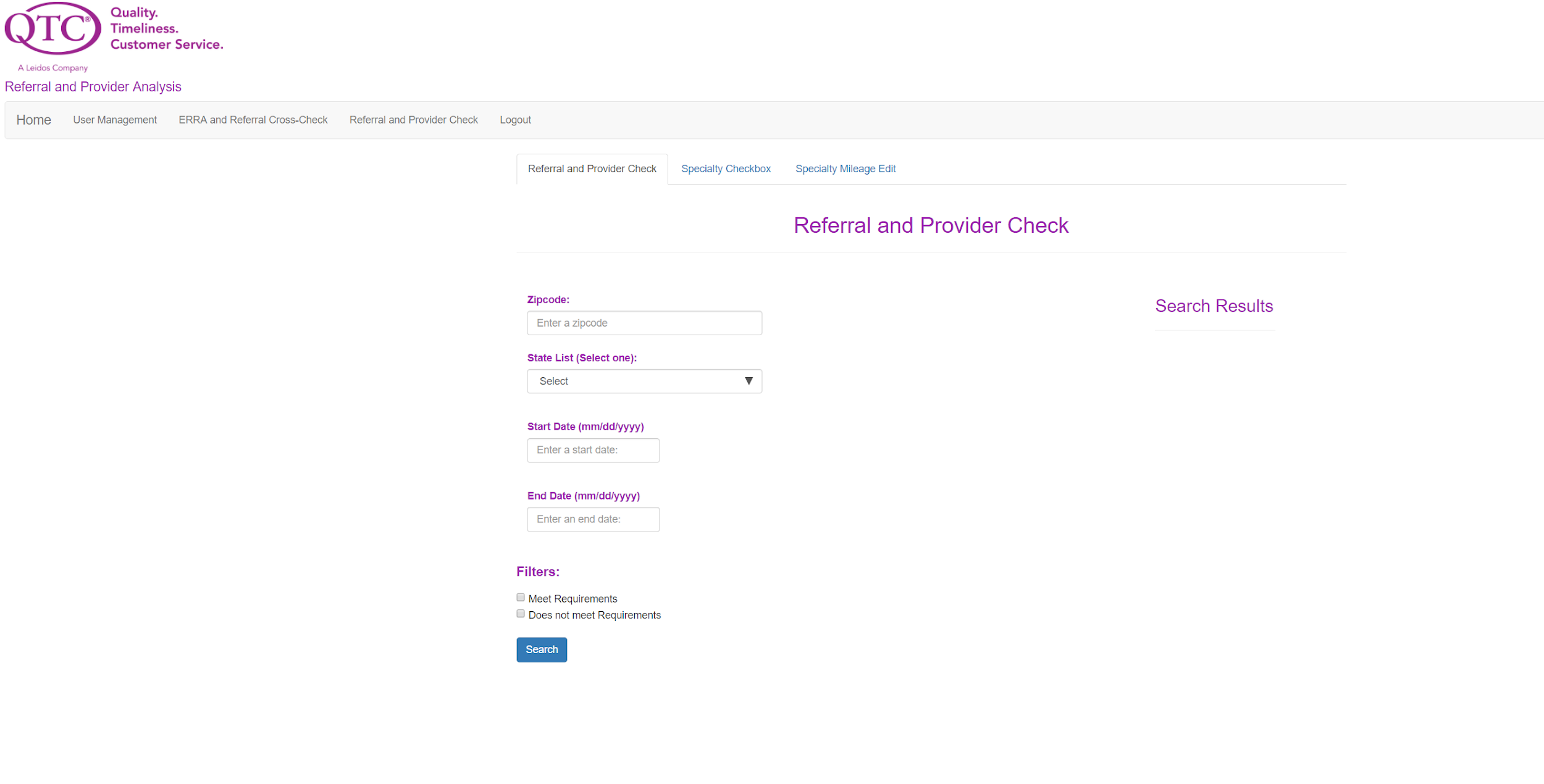
**ERRA and Referral Cross-Check**

****

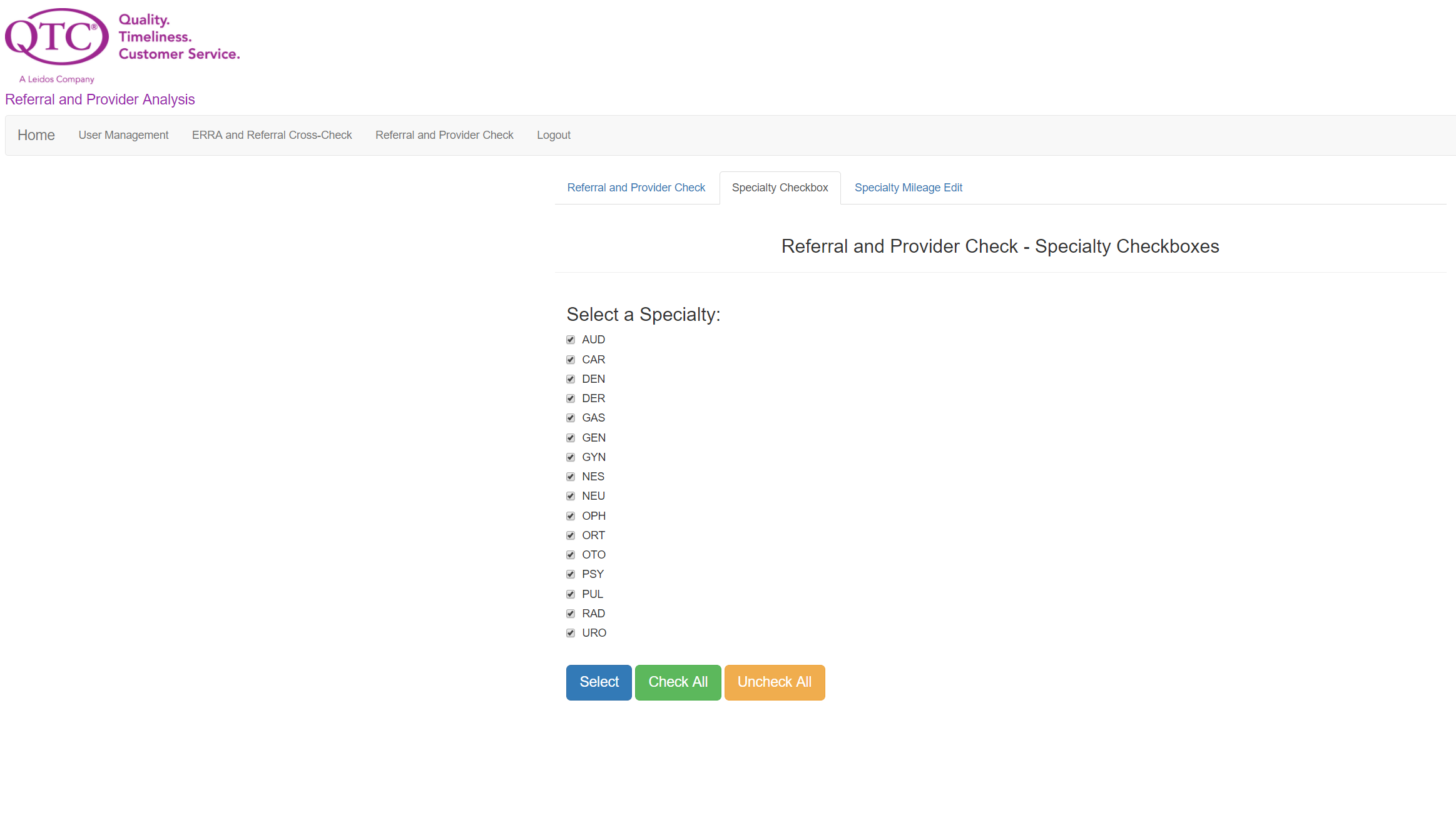
**ERRA and Referral Cross Check (Sample results)**

****

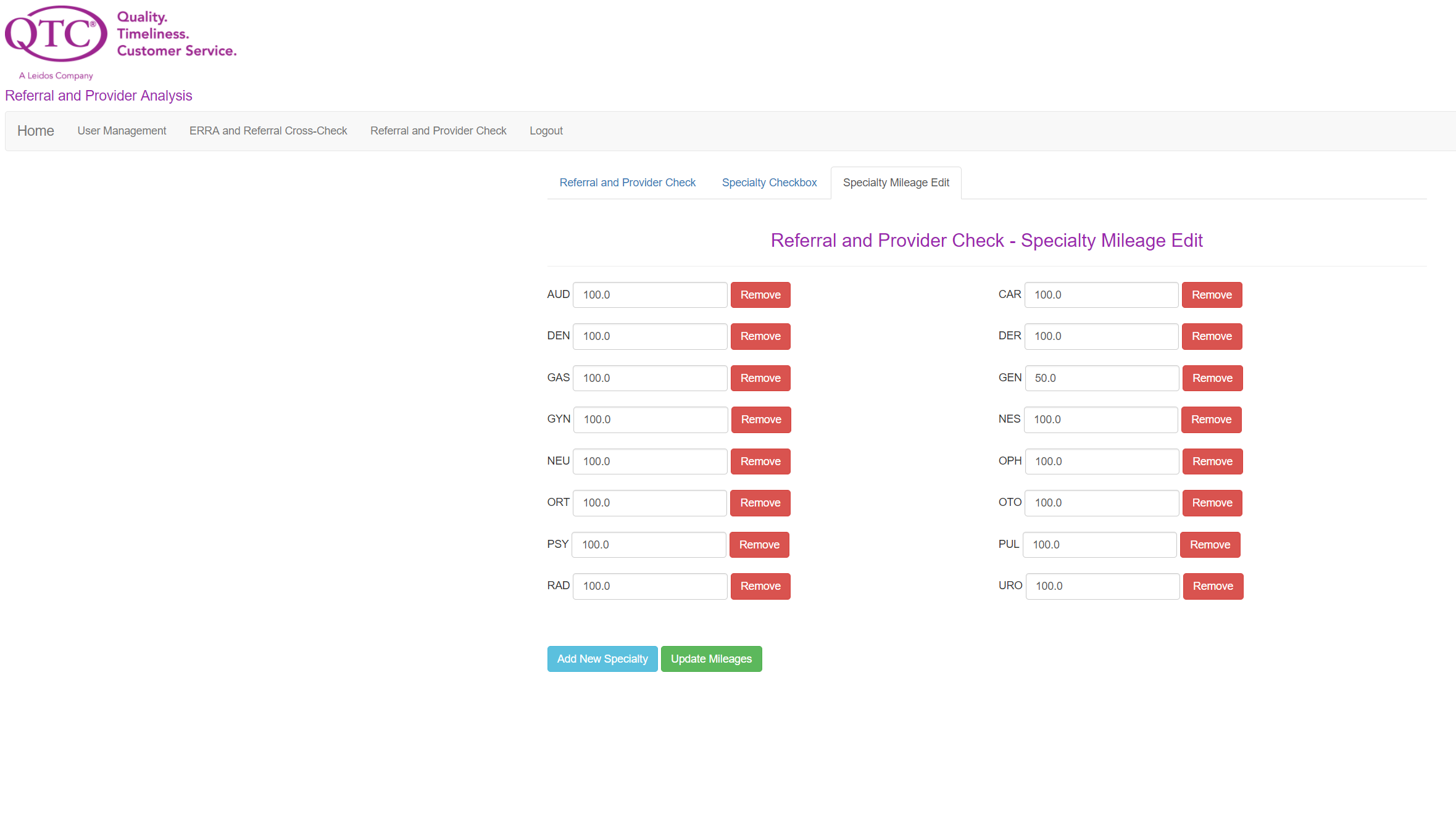
**Referral and Provider Check**



**Referral and Provider Check (Specialty Checkboxes)**

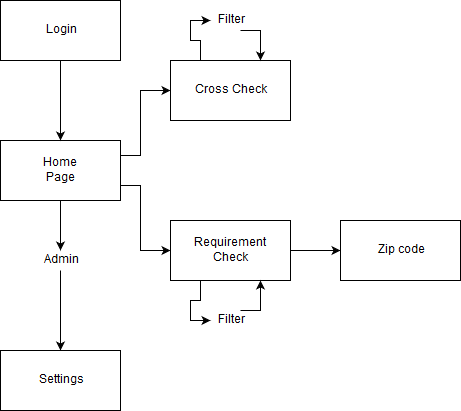
****

**Referral and Provider Check (Specialty Mileage)**

****

**8.3 User Interface Flow Model**

A discussion of screen objects and actions associated with those objects. This should include a flow diagram of the navigation between different pages.



**11. Glossary**

RE-TAT - Referral / ERRA Trending Analysis Tool

ERRA - Exam Request Routing Assistant

VARO - VA Regional Office

DOR - Date of Referral

SQL – Structured Query Language

JSP - Java Server Page

VA - Veteran Affairs

IDE – Integrated Development Environment

QTC - Quality Timeliness Customer Service

MVC - Model View Controller

UI - User Interface

**12. References**

https://commons.apache.org/proper/commons-fileupload/using.html

https://www.mindrot.org/projects/jBCrypt/

<https://github.com/Twipped/Kalendae>

https://poi.apache.org/apidocs/index.html