Technical report of DIESQNU – A Django Application Serving Electricity Quote

— IST 510 Final Project Report

Habibi, Mohammad
Wang, Hongjian
Xu, Dongpeng

5/9/2014

Contents

1 Introduction

DIESQNU is an on-line web application that serves electricity quote to users. The following functionality are provided.

- Search electricity quote by company.
- User registration.
- Log search history for registered user.
- Registered use can subscribe specific query.
- When quote is updated, notify the registered users.

The name of our application — DIESQNU — is derived from its supporting platform and functionality. This application is a *D*jango based server that provides *I*nformation *E*xtraction, *S*erves user *Q*ueries, and Notifies query Updates. Speaking of the supporting platform, DIESQNU is built over the following tools.

- Django framework is an easy to use high-level python web framework that encourages rapid development and clean, pragmatic design.

- MySQL to manage the electricity data we extract from the PDF file.
- Implement an SMTP client to send email to users as notification.

The structure of this report will be organized as follows. The next section will cover all the functionality that is implemented in our web application. In section 3, we will show the technical details of some challenging task. We will conclude this report in section 4.

2 Functionality Implemented

This section we will introduce the system functionality that we implemented.

2.1 Query Quote

User can query the electricity company's quote in our system. User is required to input the name (or partial name) as query keywords. Then, a fuzzy matching is called to search the database for possible match. In the results page, we will give the total number of results first, followed by the detailed electricity quote of each company. Notice that there are four kinds of different validation time for the quote.

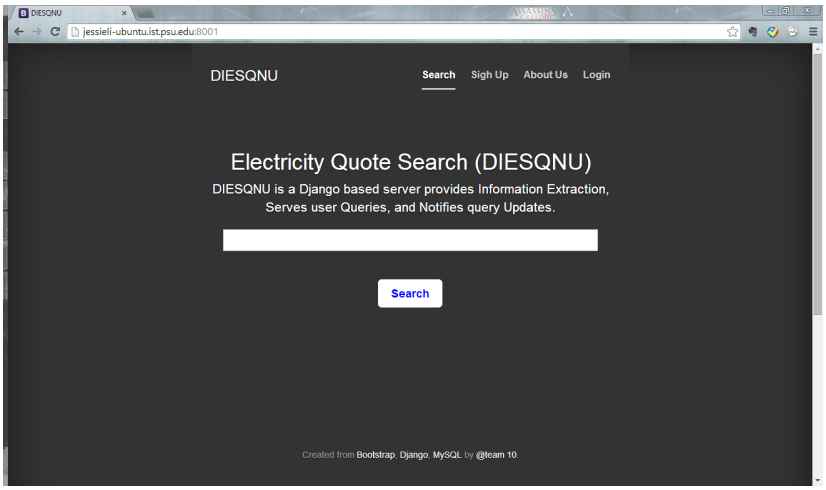

Figure 1: Query page UI design.

2.2 User Registration

User can registers in our web application and gets additional services such as the query change notification.

During the registration, system will check the availability of user name first, because no duplicated user names are allowed. Each entry in the sign-up table should be filled with correct information. Any mistakes in those fields will be notified to user.

2.3 User Login

Registered user can login to our applications. Wrong passwords or in-valid user name will raise an error message.

After user logging in the web application, the sign in link will be replaced by a sign out link. There will be an additional welcome panel on the upper right conner of the page to notify the identity of logged user.

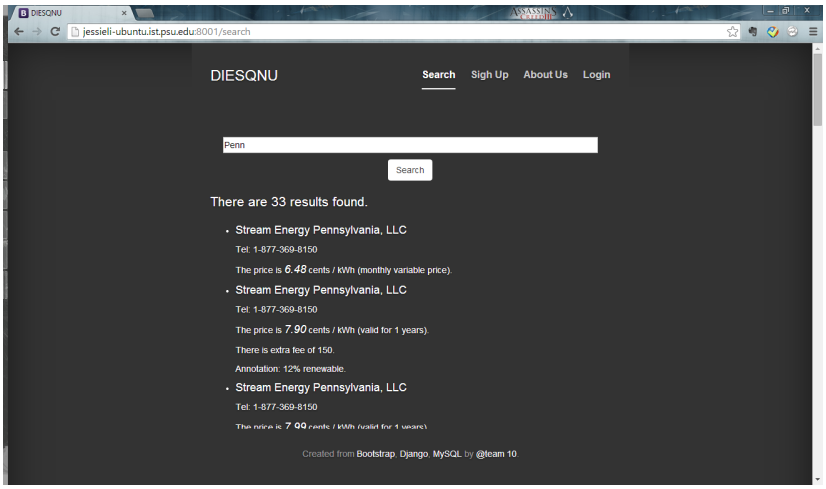

Figure 2: Query results for keyword "Penn".

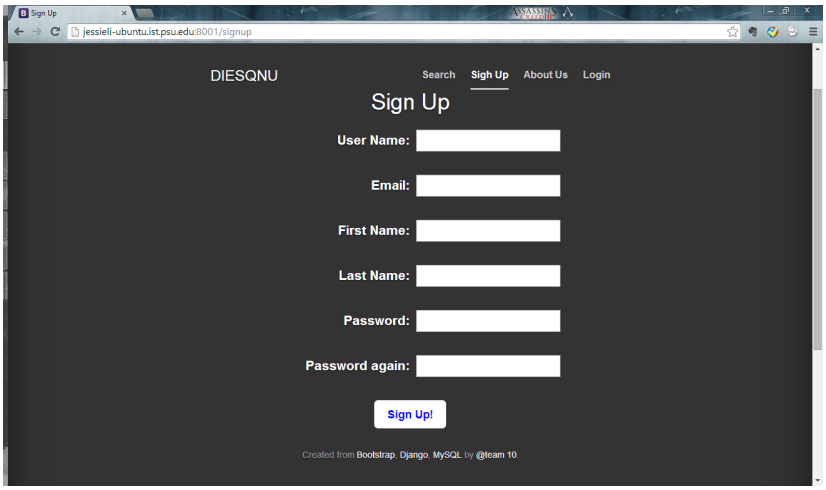

Figure 3: Sign up page for user.

2.4 Query Log and Notification

When user is logged into the web application, there is a link to the query log of the logged user in welcome panel. The query log page enables user to register query for change notification, as shown in Fig 7.

In the query log page, we show the query string, time of that query, number of results and the registered status of that query. If the query is registered, the checkbox of this query is checked, and there is a link that allows us to unregister that query. On the other hand, if the query is not registered, we can check the unchecked box in front of it, and then using register query button to register it.

Notice that there is a "check update" button in the query log page. Our server will check the query changes periodically on the background. Here, for the sake of demonstration, we add a button to manually call this function. After calling the update check, the user can get the email notification, shown in Fig 8.

2.5 Administration Console

The Django framework provides us a free administration console, where we can directly see all the data in database and manipulate those data. A snapshot of this administration console is shown in

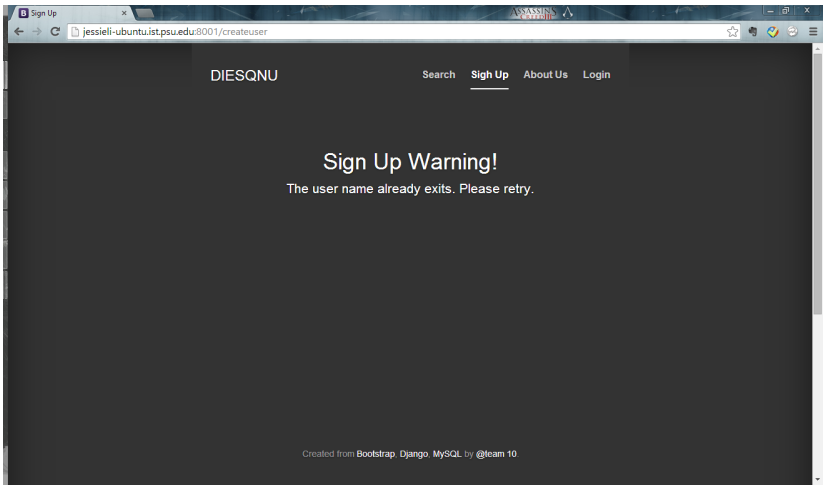

Figure 4: Sign up page result, when an error happens.

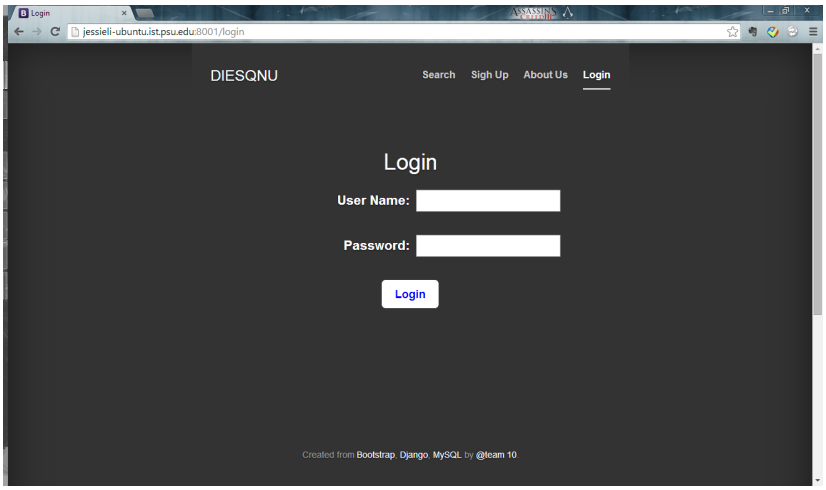

Figure 5: Log in page for user.

Fig 9.

3 Technical Details

3.1 Crawling

add Dongpeng's part from Google doc.

3.2 Identifying Changes and Sending Email Notification

add Mohammad's part from Google doc.

4 Conclusion

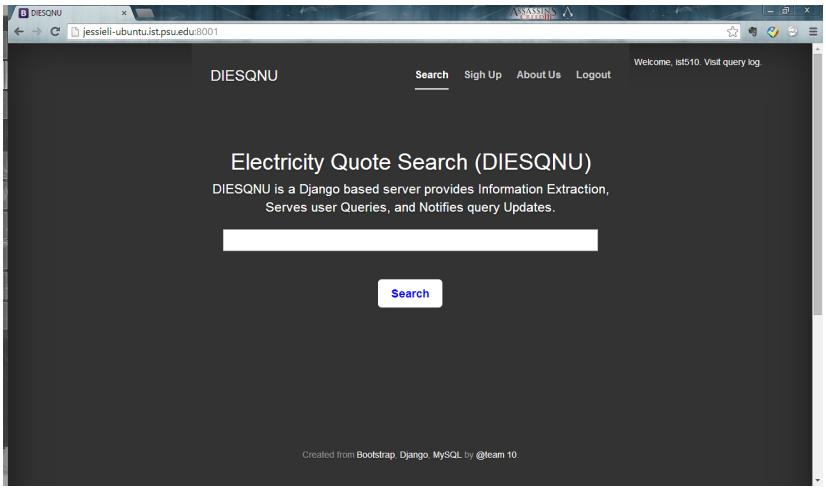

Figure 6: After user logged in application, there will be a welcome panel in the right conner.

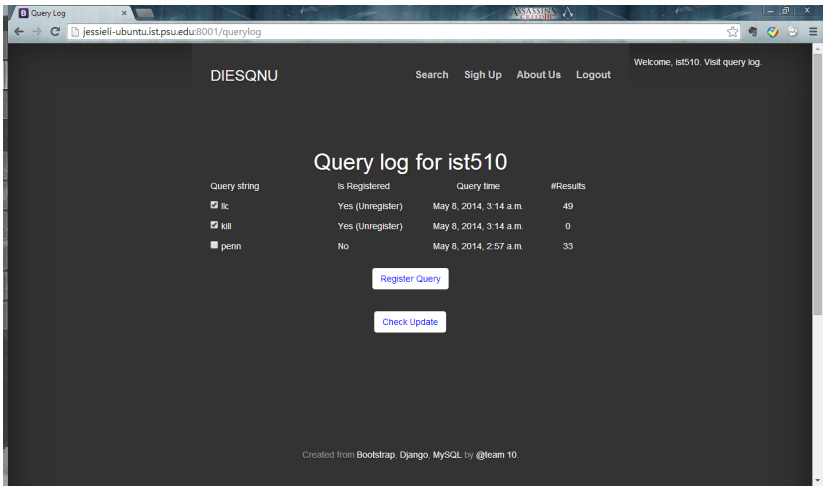

Figure 7: The query log page, user can mange the query history from here.

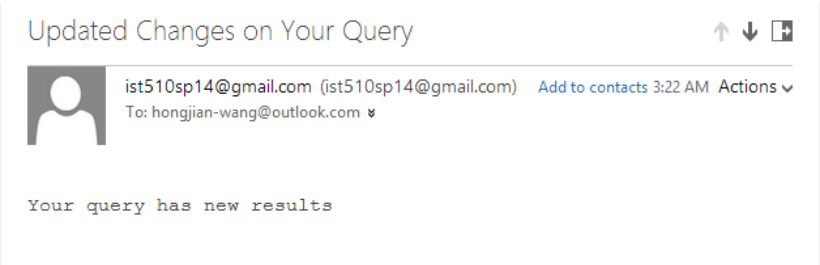

Figure 8: User can get email of query change notification.

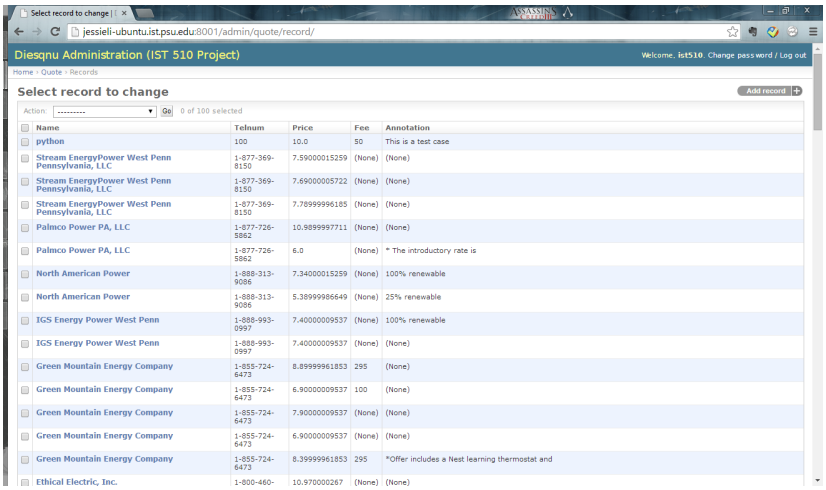

Figure 9: Administration console.