**A: Description**

**1)List of Classes:**

In my program, there are eight classes:

•**Name.java**

In this class, there are two fields firstName and lastName. This class is used to store the full name of one entity.

•**Address.java**

In this class, there contains all fields of an address. This class is used to store the address information of one entity.

•**AddressEntry.java**

This is an entity class, all the fields of one person are

defined in it.

**•AddressBook.java**

In this class, firstly there is a field called addressEntryList. This is a ArrayList and each addressEntry was stored in it. I choose the structure ArrayList because it is convenience and effective in search, just calling the get(i) method which already provided (i is the index of the entry). Also add and remove operations of ArrayList are very quick.

The second field called myaddressEntryListModel, which is used to bound to Jlist to display on the MainWindow.

The last field called comparator which is a instance of the anonymous inner class—Comparator. This anonymous inner class override the method compare and let the entries sorted in alphabetic order by the person's last name (ignore case).

Besides the getters and setters, there are eight methods in this class. Including readFromDB, addNewToDB, updateInDB, deleteFromDB, sorted, search, remove, and add.

Read, add, update and delete operations of database implemented with some sql statements. The method sorted call the sort method in class Collections and use the override comparator. The time complexity of sorted is O(n2). The time complexity of search and add and remove are all O(n).

•**MainWindow.java**

This class displays all the entries of the AddressBook, and provides four buttons: “New” button for add an entry, “Remove” button for delete an entry, “Update” button for update the information of one entry, and “Search” button for search by last name.

•**MyJdialog.java**

This is a dialog interactive with users. It will ask user to enter some information and when the user hit save, these information will save to the application, otherwise, if the user hit quit, the information will lost.

•**SpringUtilities.java**

This class provides utility methods for creating form- or grid-style layouts with SpringLayout.

**•AddressBookApplication.java**

This is the main function. In this class, just calling the getFrame method of class MainWindow.

**2) State of System:** Full complete.

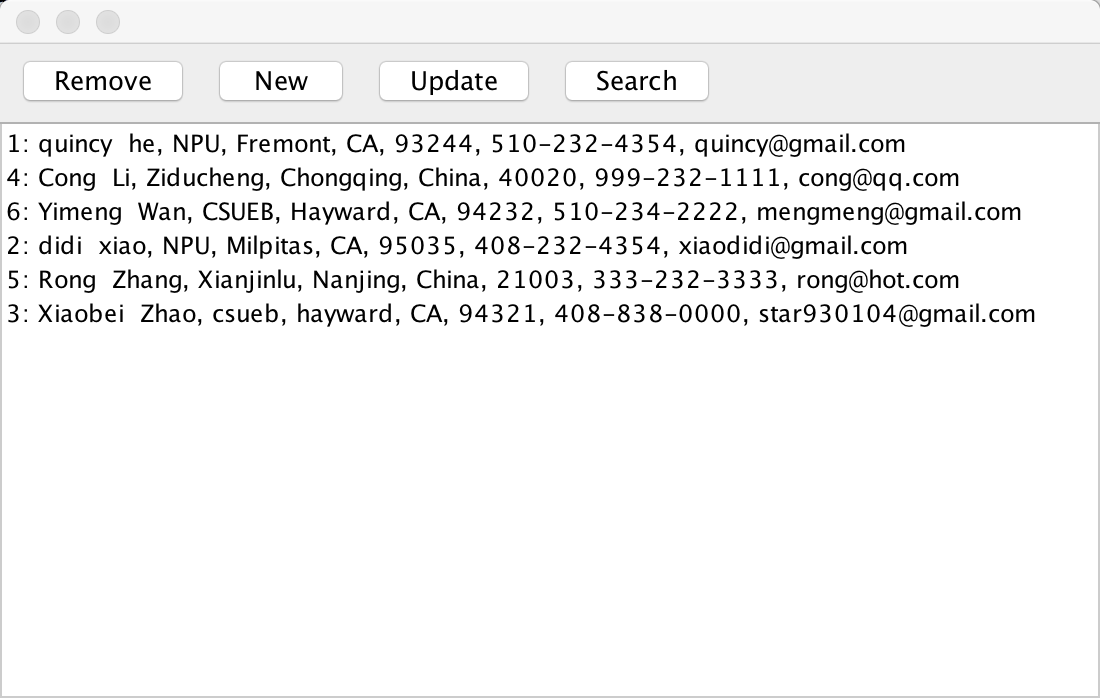
**B: Javadoc URL:** <https://proj2-javadoc-xiaobeizhao.herokuapp.com/index.html>

**C: UML/Design:**

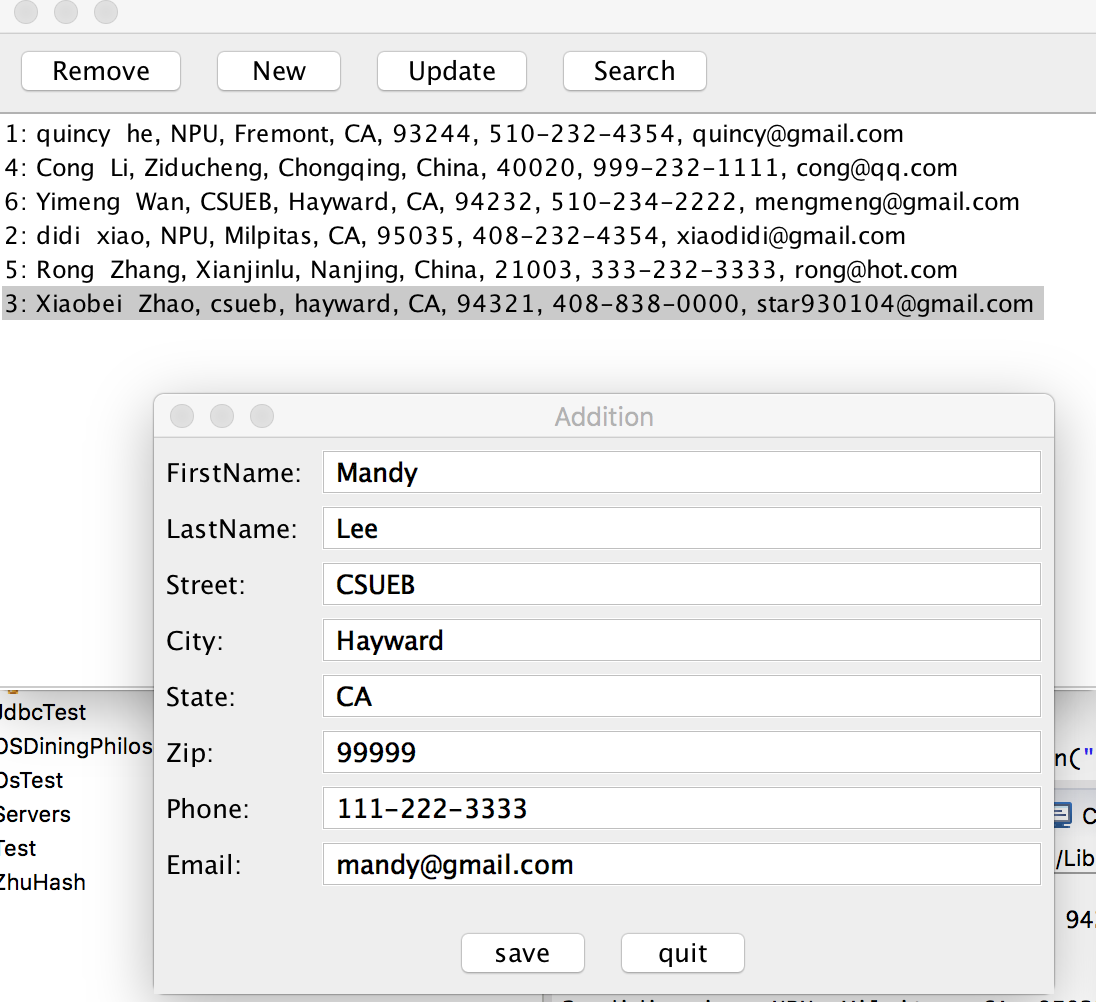


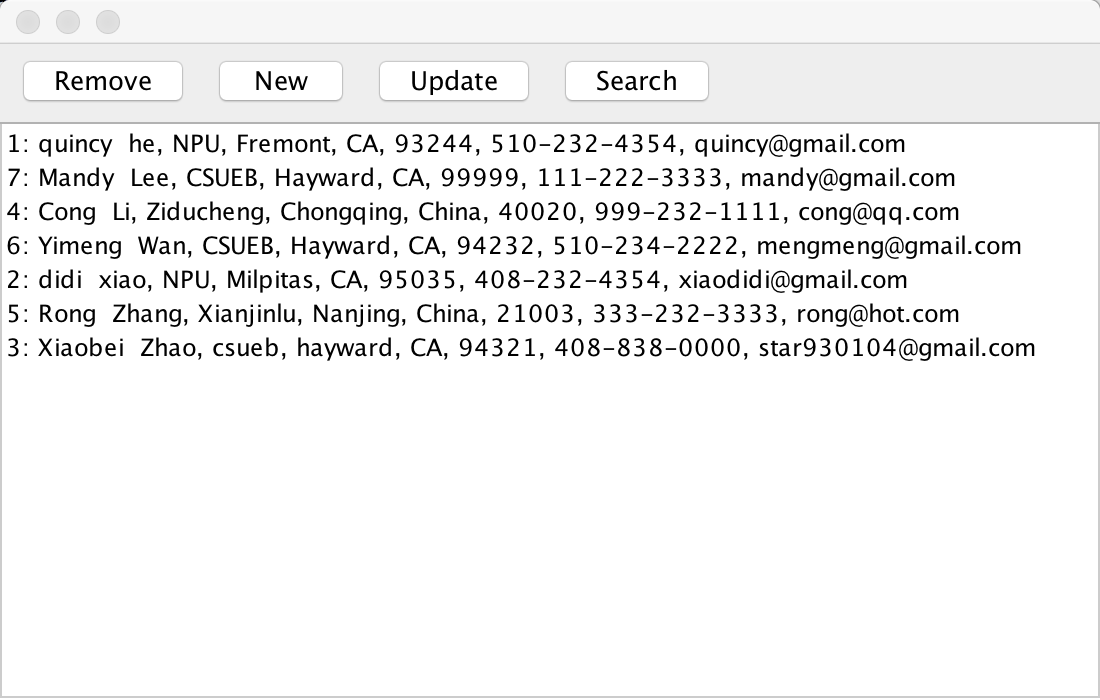
**D: Working Screenshots:**

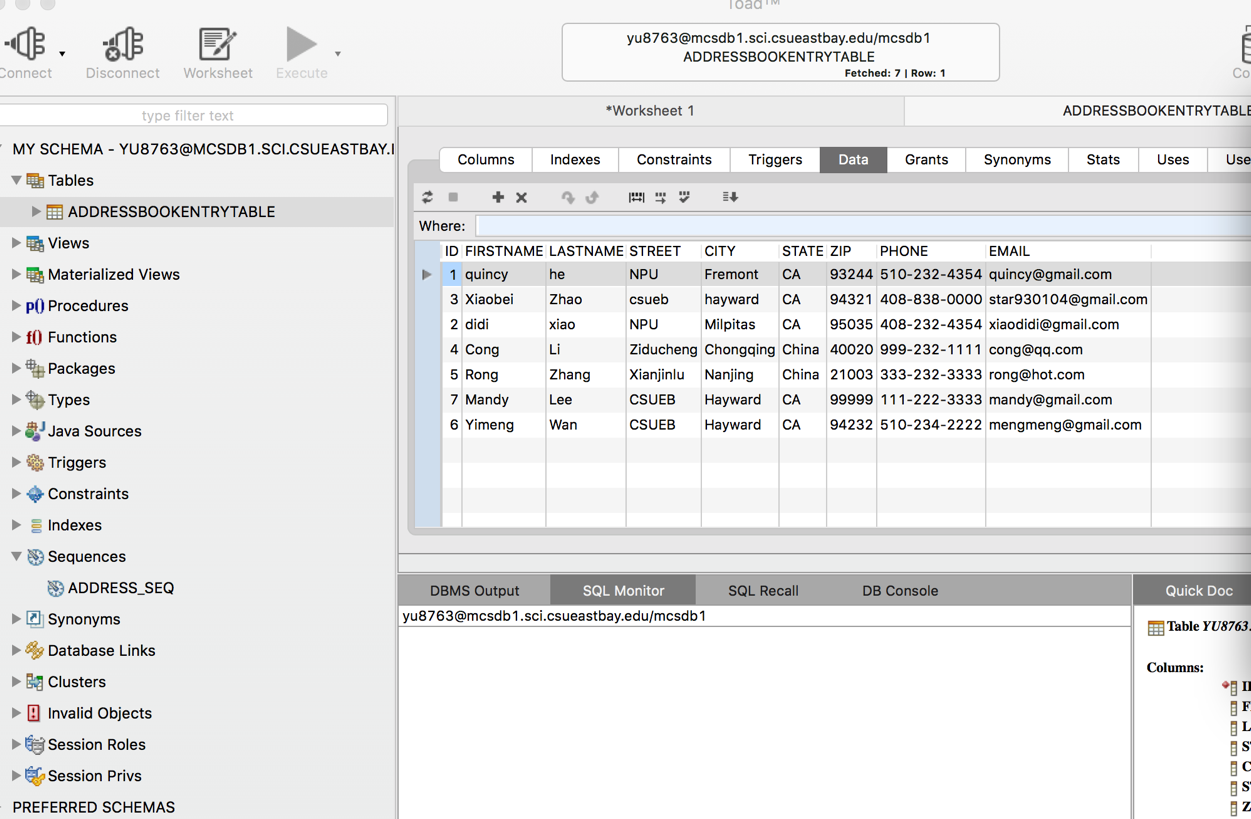
1. Read in entries from database.



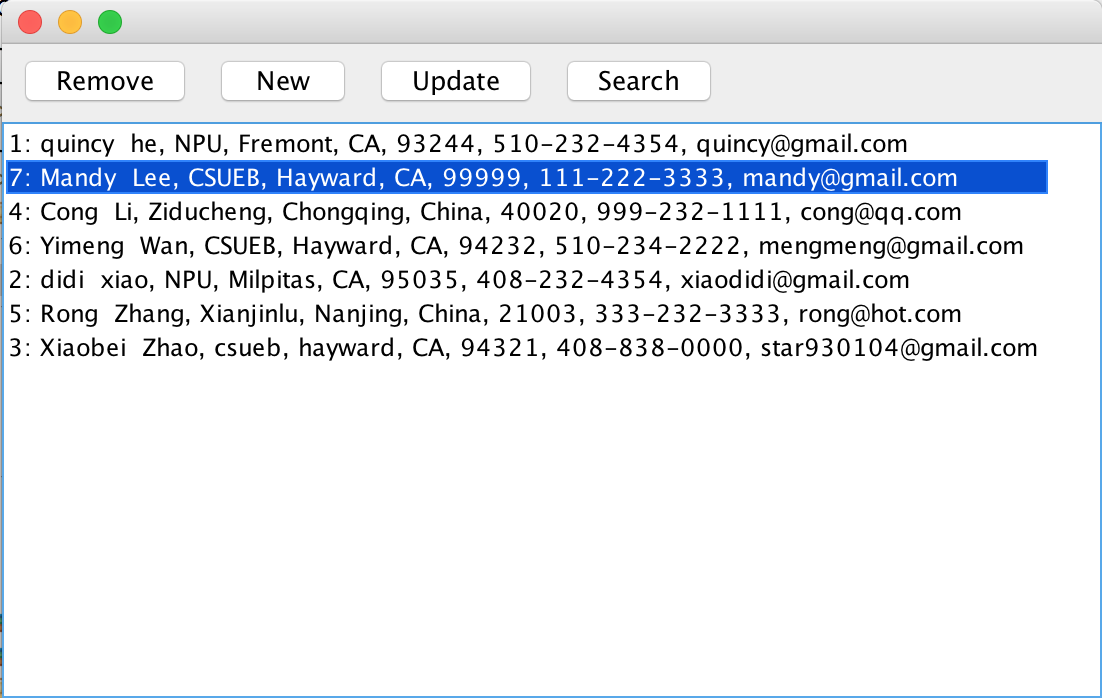
1. immediately addition a new AddressEntry.

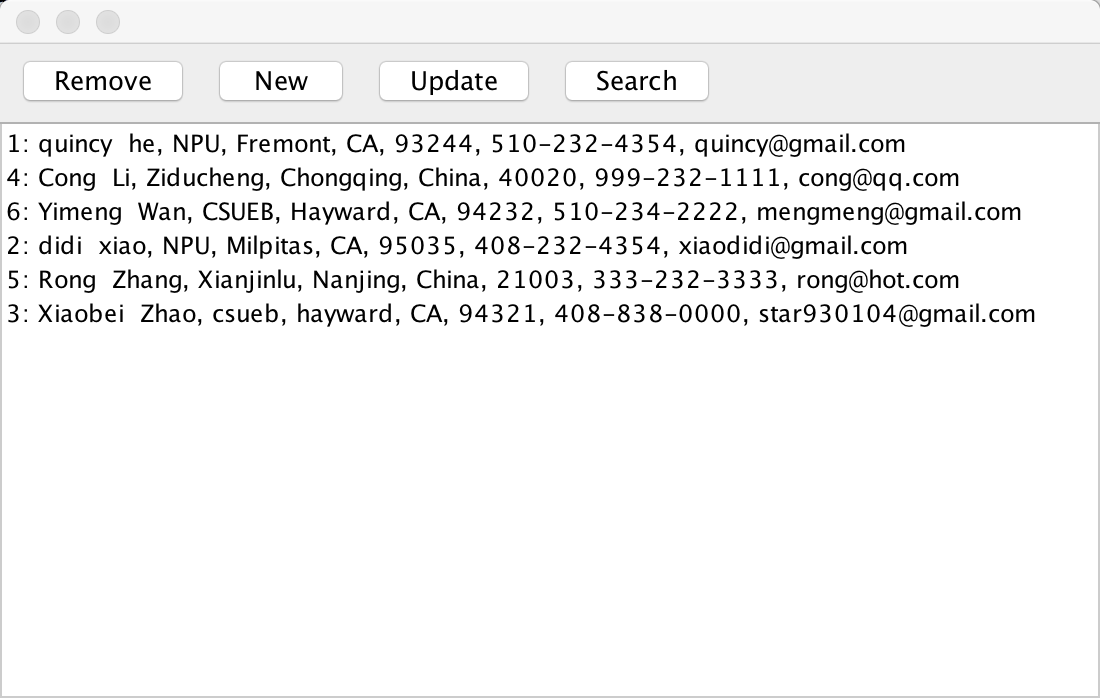


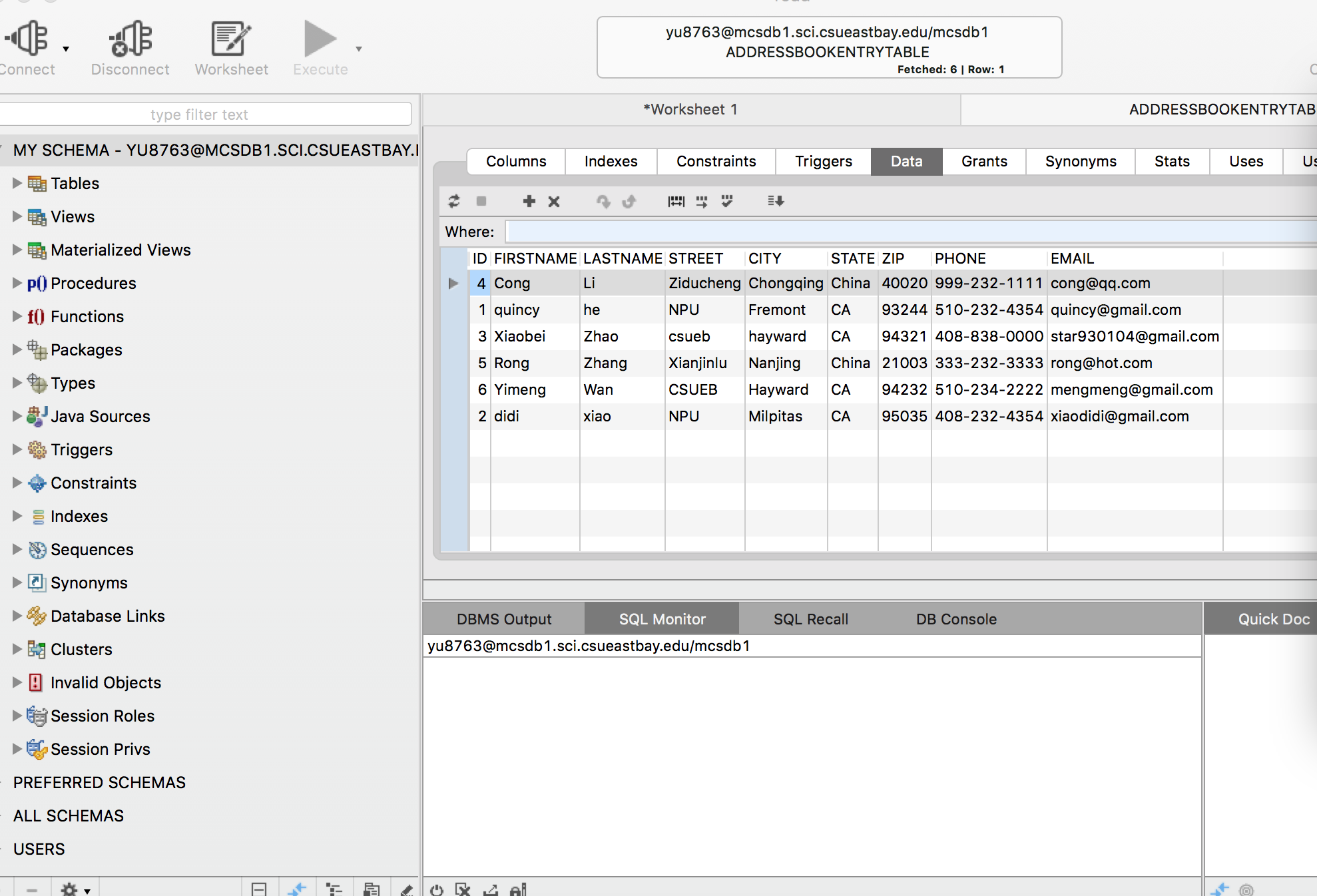




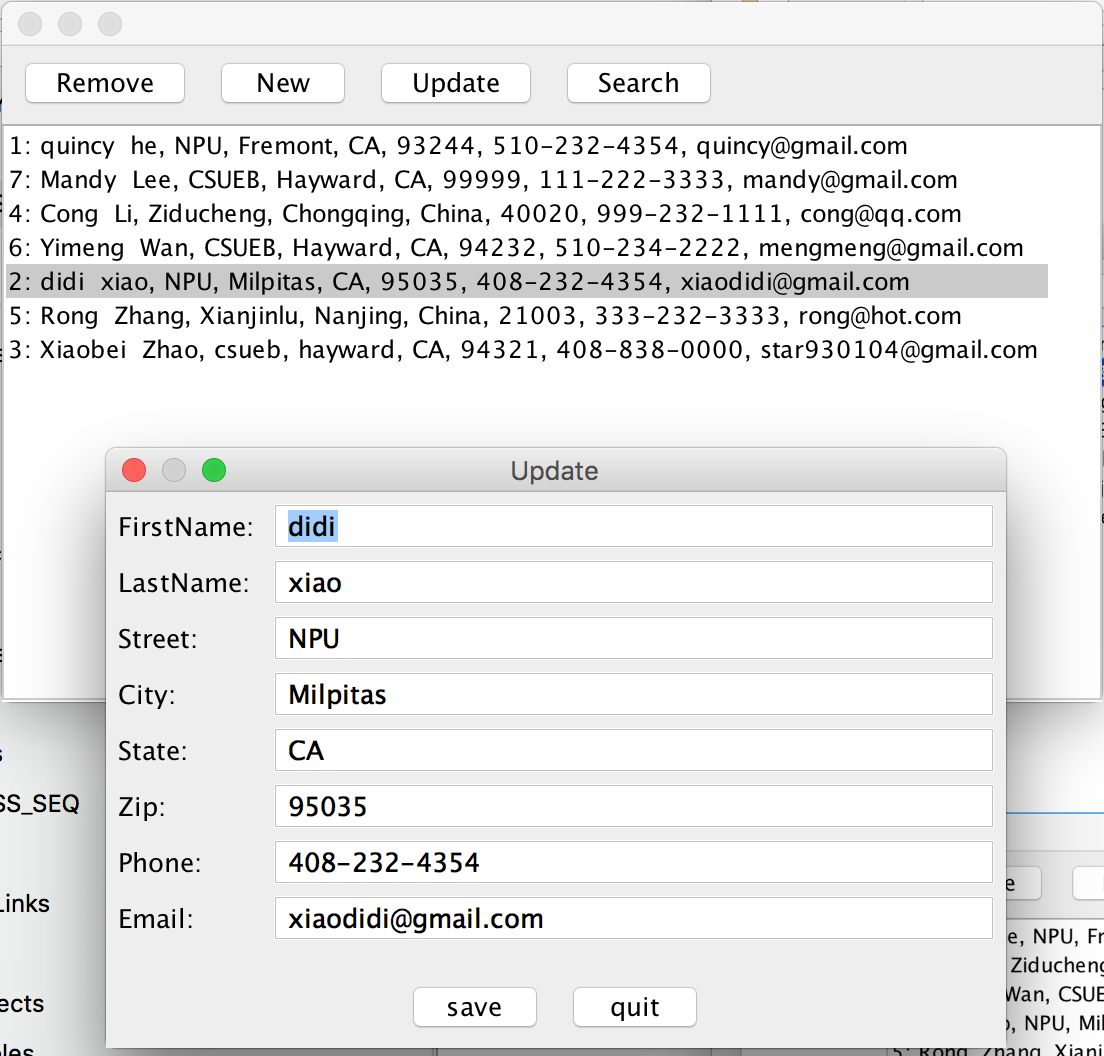
3) immediately removal of a entry.

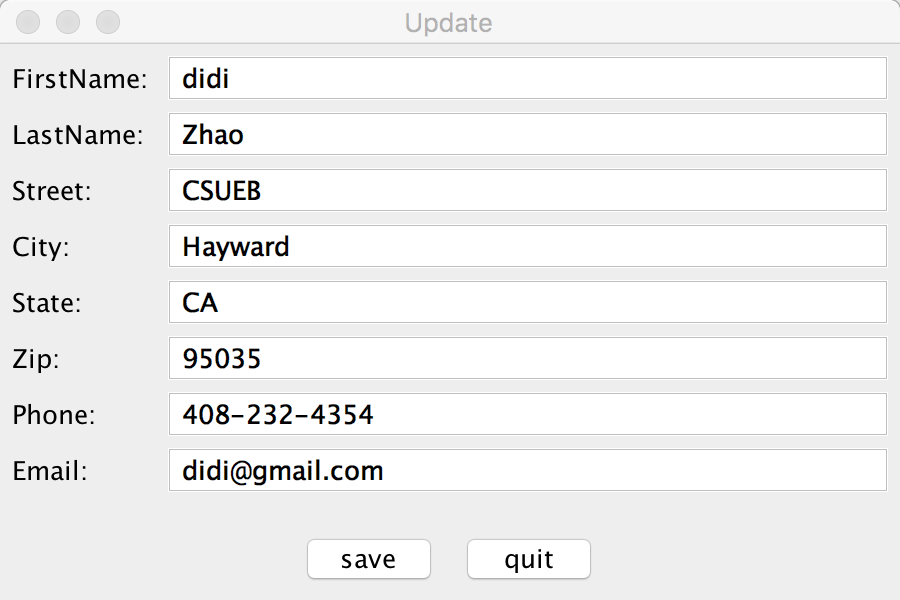


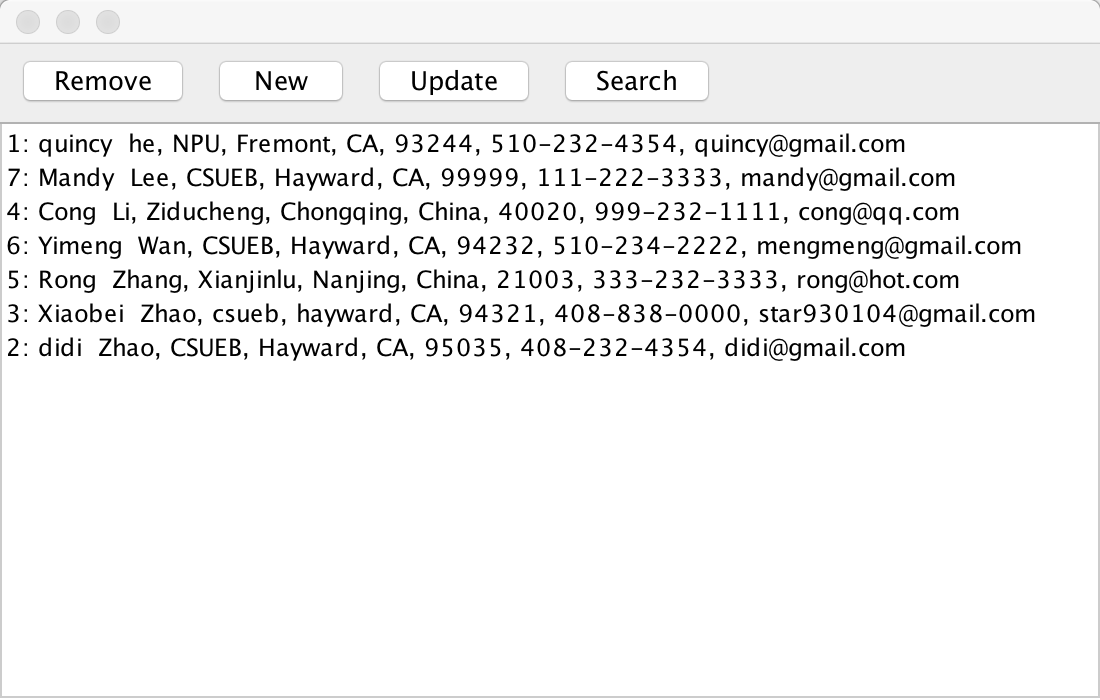


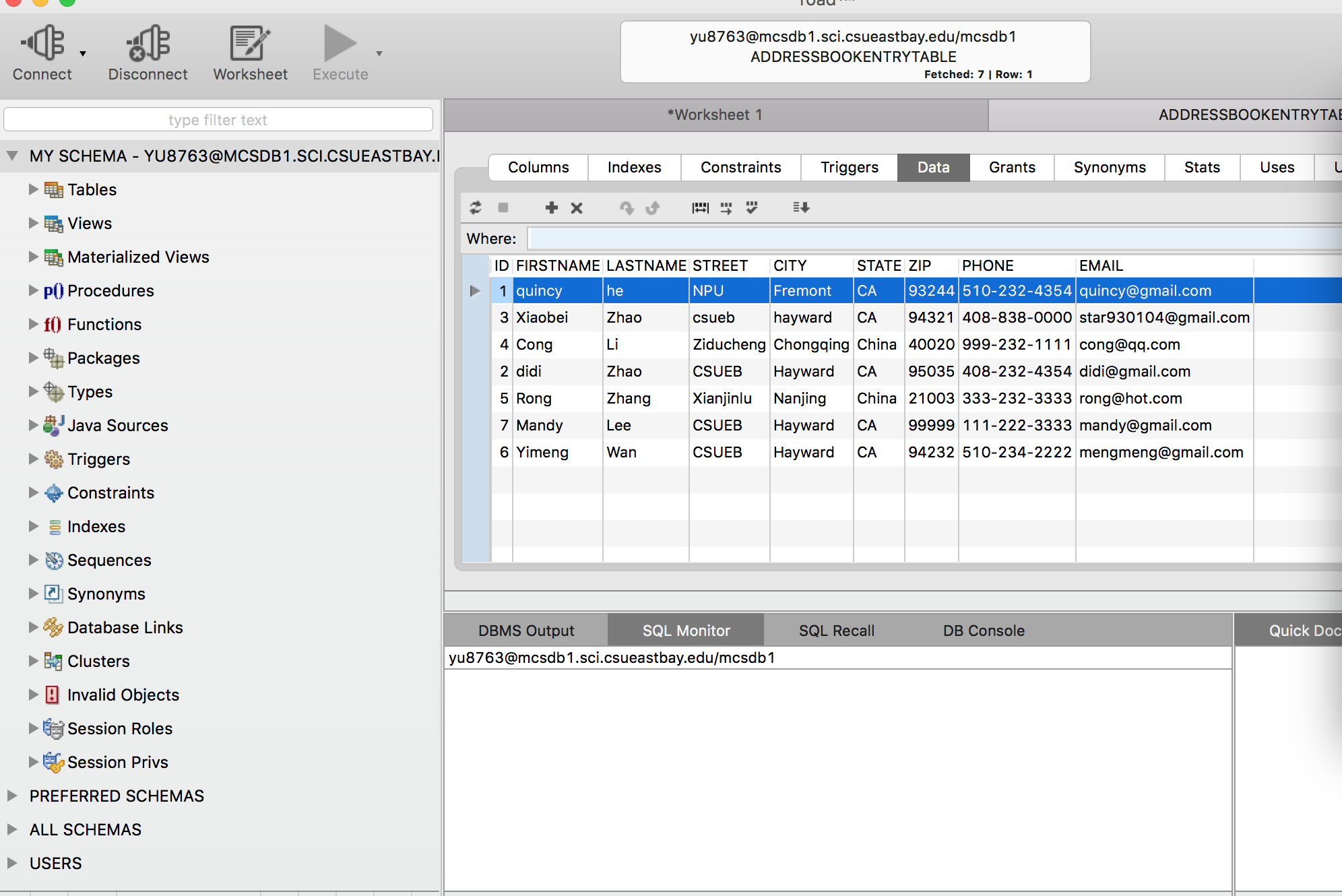


4) immediately update.

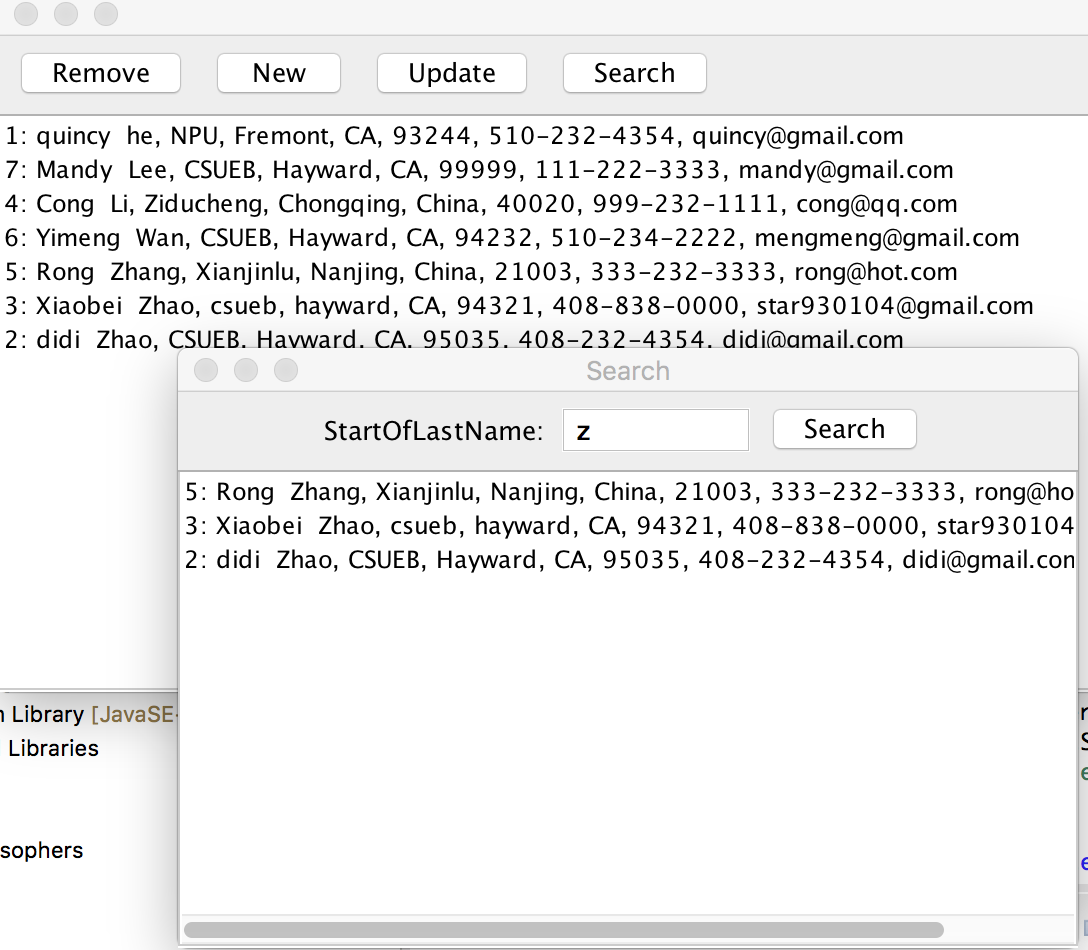








5) find.



**E. Screen shots showing of the BitBucket**

