2022

RCM Information Suite Instruction Manual

A QUICK START GUIDE FOR THE RCM INFORMATION SUITE

ISAIAH C BOWERS

COPYRIGHT © ISAIAH C BOWERS | All Rights Reserved

Table of Contents

Running the Program 3
Machine Learning – READ ME! 4
Why is the Machine Learning Aspect Useful? 4
How a User Might Test The Machine Learning Portion of the Application 5
How Remark Codes Can be Predicted by Search Phrase: 5
How the Application can Show Remark Codes Which Have Been Most Used With Each Other 8
Typical Use 11
Searching for Information 11
Making Suggestions 11
Unique to Remark Code Tab 12
Unique to Payer Tab 13
Unique to Provider Tab 13
Administrator Functions 14
Admin Information Search 14
Admin Menu 14
Charts 16
Code Summary – Why You Should Hire Me 17

Running the Program

To run the program, follow the following steps:

- Ensure that you are using a Windows operating system.
- Navigate to the application folder – the same folder which contained this document upon download – called "RCMInformationSuite_jar."
  - ✓ Double-Click the "RunRCMInformationSuite.bat" file in this folder.
    - The command line terminal – typically a black window with white letters – will open. This is normal and this window must remain open for the duration of the use of this application.
    - Several seconds after the command line terminal has opened, the RCM Information Suite will open.

Machine Learning – READ ME!

One of the best parts about this application is that it uses machine learning (reinforced learning in particular) to assist the user in using the correct billing codes – also known as remark codes – in the healthcare industry.

The RCM Information Suite does this by:

- Keeping a list of all remark codes searched by the user within the last five minutes.
- Whenever a remark code is copied to the clipboard or added to a list, all search phrases on the list gain one "association point" with the selected remark code and the list is immediately cleared.
  - ✓ The next time any of these phrases is searched, the application will return all remark codes which are associated with the phrase, followed by any remark code whose code or description contained the search phrase.
- Additionally, the application keeps track of which remark codes are used together (done by adding the remark codes to a list view and then copying them all to the clipboard at once, separated by commas).
  - ✓ Any time a remark code is selected from the "Results" table, the "Most Commonly Used With Your Selection" table is populated with the appropriate remark codes.

Why is the Machine Learning Aspect Useful?

The machine learning aspect of this application is useful because there is a standardized definition for each remark code – but insurance companies often use their own made-up definitions. A payment poster then must decipher which remark code the insurance company was actually trying to indicate.

By storing which search phrases are associated with which remark codes and ordering the remark codes by which have been most used with the search phrase, the stress incurred by the payment posters is much less.

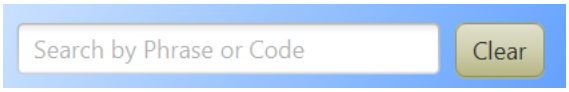
The same is true about keeping track of and allowing the user to see which remark codes are most commonly used with which other remark codes.

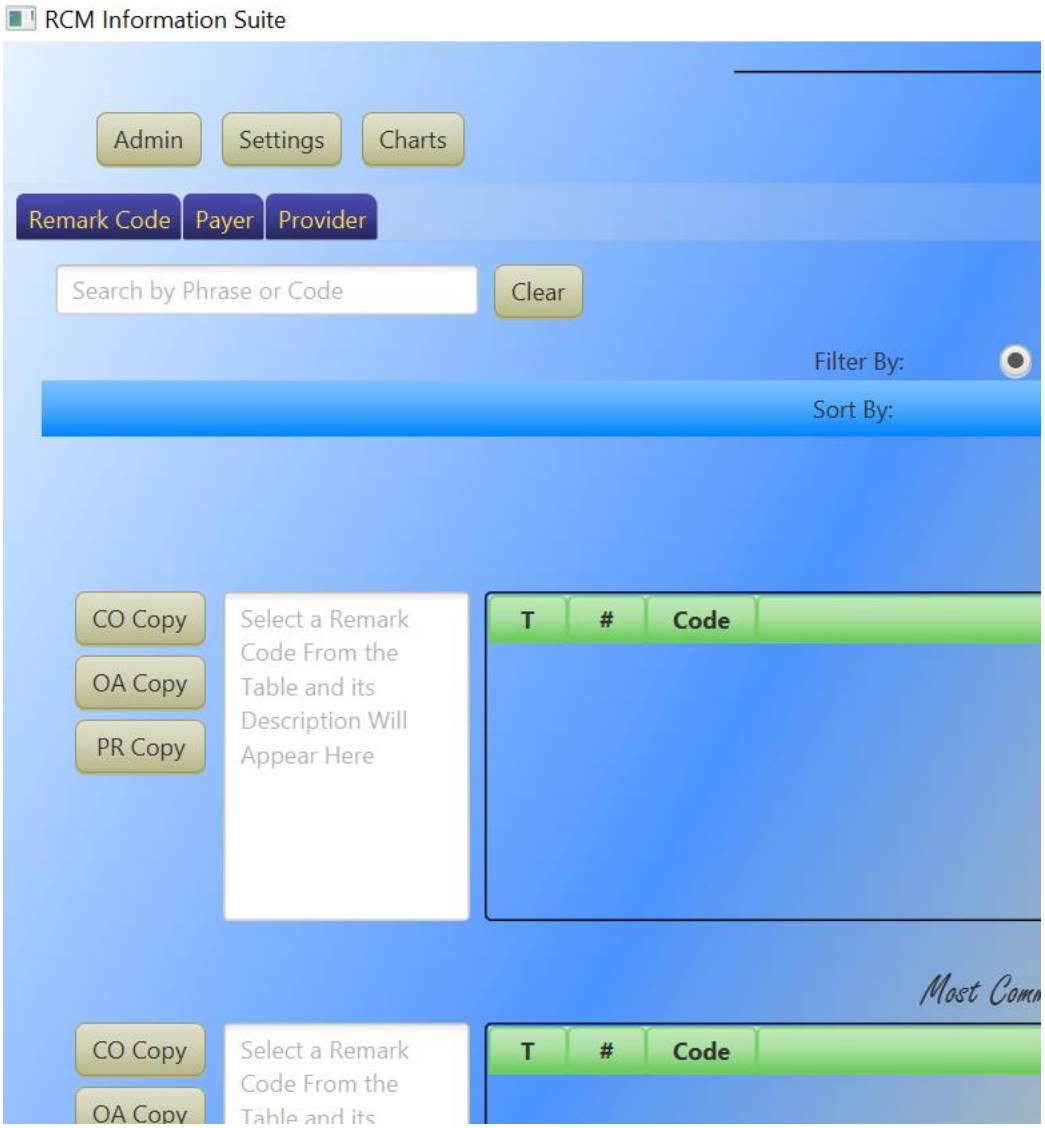
How a User Might Test The Machine Learning Portion of the Application

How Remark Codes Can be Predicted by Search Phrase:

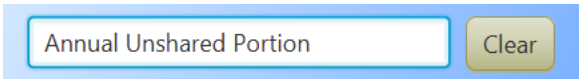
I will list the steps, enumerated as 1., 2., 3., etc. and include pictures for some of those steps.

1. Understand that “Annual Unshared Portion” is commonly used by healthcare payers to refer to a “Deductible.”
2. Start the application using the instructions in the “User Guide” portion of this paper.
3. Click in the search text field in the “Remark Code” tab in the “Information Search” menu – the first menu to which the application opens – located in the top left corner of the tab.

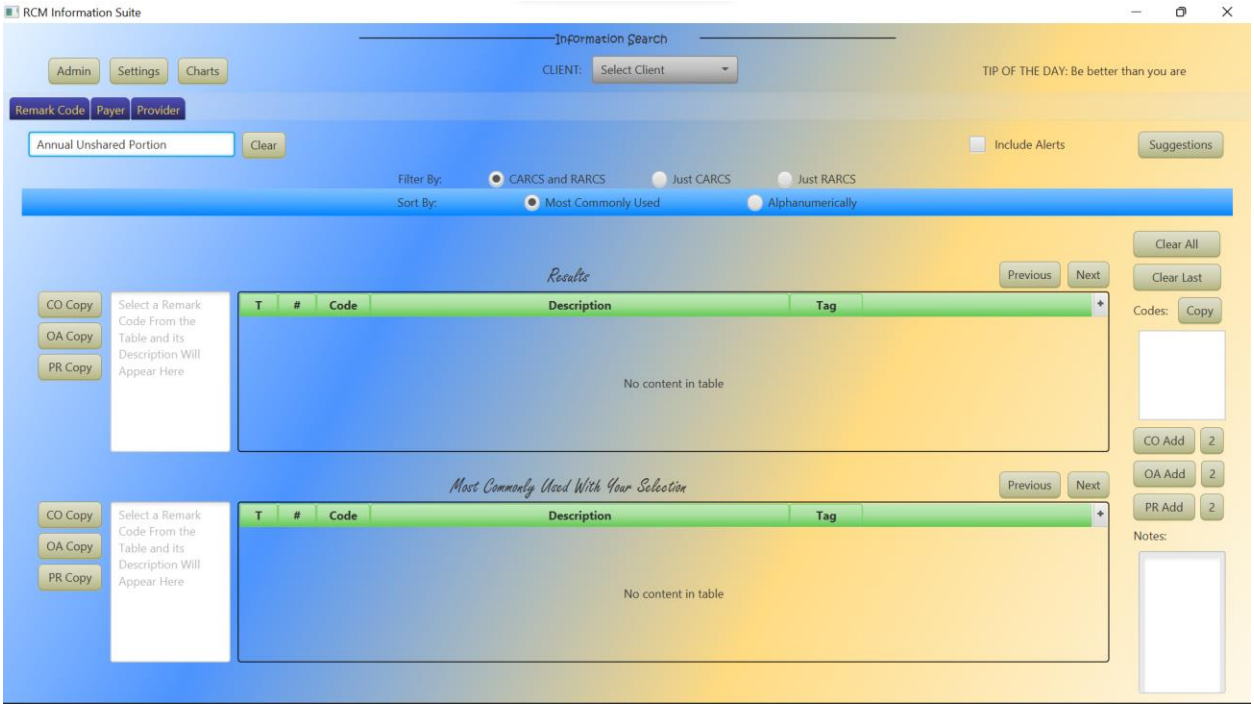




4. Type “Annual Unshared Portion” into the text field.

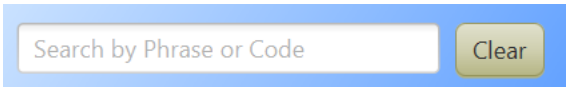


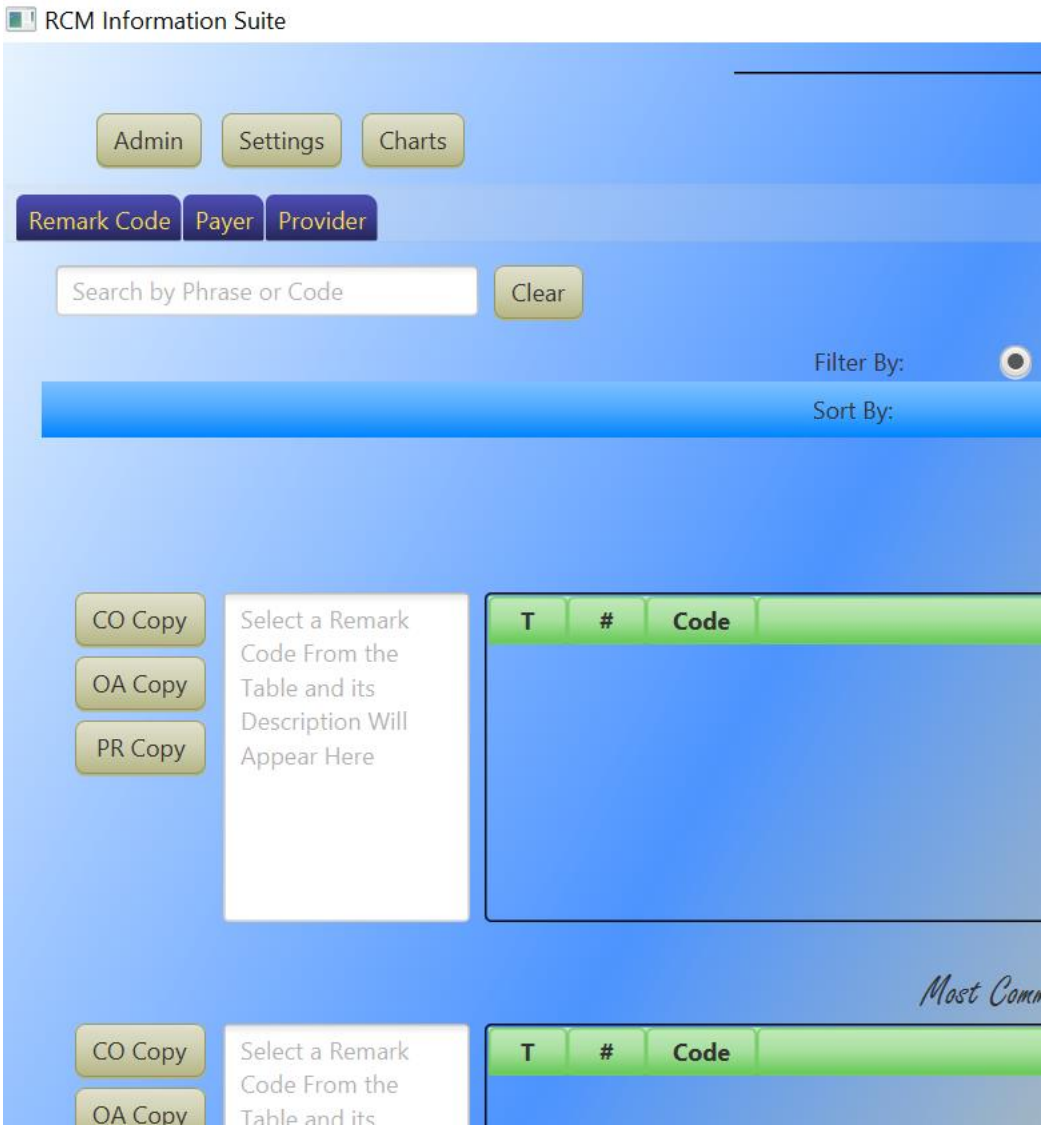
5. Hit the “Enter” or “Return” key on the keyboard.
6. Understand that nothing came up for this search. If something did come up for this search, either choose a different phrase to search – it can be anything – or, for future steps, be ready to choose a remark code which did not appear in the search results



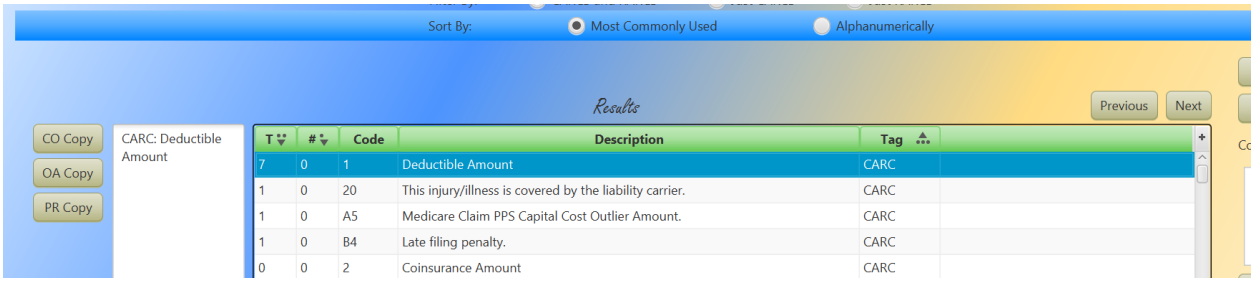
7. Click on the “Clear” button on the “Remark Code” tab on the “Information Search” menu – the first menu to which the application opens – located next to the previously used search text field.

8. Click in the search text field in the “Remark Code” tab in the “Information Search” menu – the first menu to which the application opens – located in the top left corner of the tab.

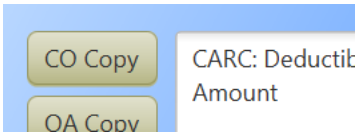




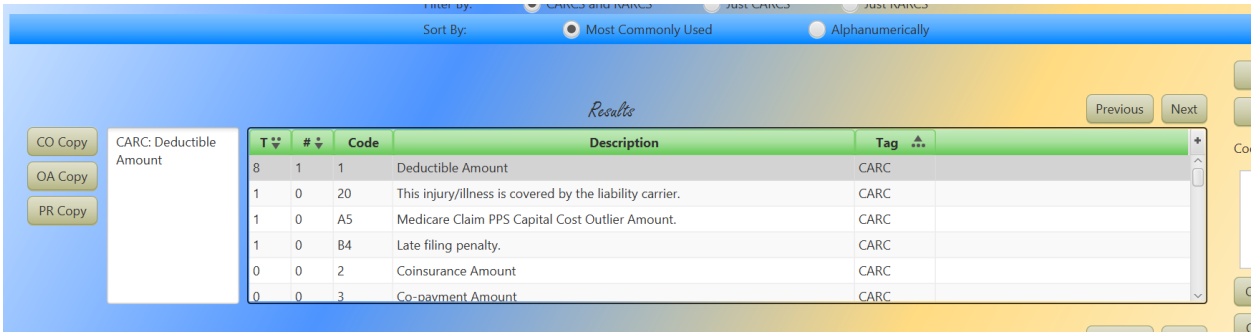
9. Hit the “Enter” or “Return” key on the keyboard.
10. Select “Deductible Amount” or any other remark code from the results list by clicking on it within the table.



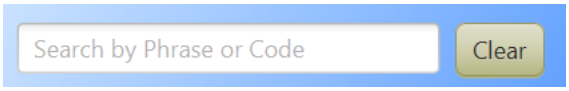
11. Click any of the copy buttons to the left of the “Results” table. The CO Copy button will work fine.



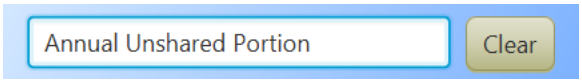
12. Understand that the “T” and “#” column amounts have increased – by how much will depend on the exact actions taken by the user



13. Click in the search text field in the “Remark Code” tab in the “Information Search” menu – the first menu to which the application opens – located in the top left corner of the tab.



14. Type “Annual Unshared Portion” or whichever search phrase was previously used and which returned no/limited results into the text field.

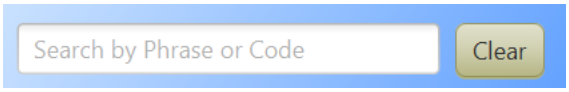


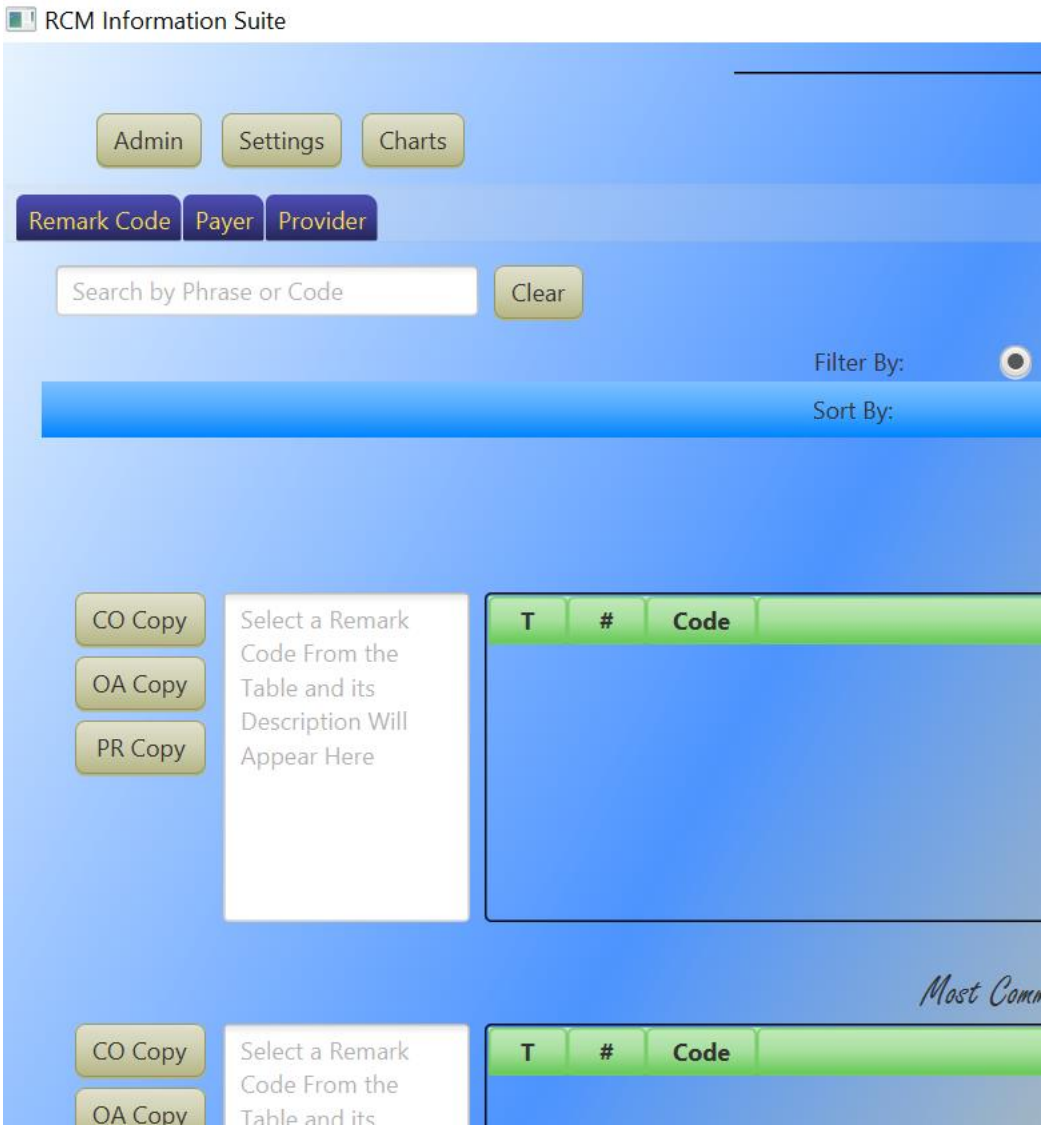
15. Hit the “Enter” or “Return” key on the keyboard.
16. Understand that “Deductible Amount,” or whichever remark code the user chose in previous steps, should appear in the “Results” table
17. If the user wishes to see that the association points increase appropriately, they may repeat the steps except select a remark code which already appears for their initial search.

The accuracy of this non-descriptive method will depend on how frequently the application is used; because this is a reinforced learning application, it will initially have a high misfire rate which should decrease significantly with use.

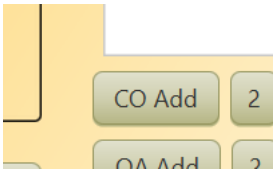
How the Application can Show Remark Codes Which Have Been Most Used With Each Other

1. Click in the search text field in the “Remark Code” tab in the “Information Search” menu – the first menu to which the application opens – located in the top left corner of the tab.

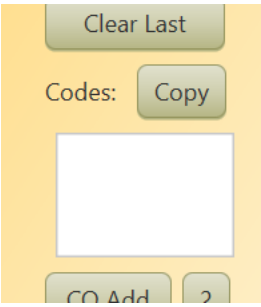




2. Hit the “Enter” or “Return” key on the keyboard.
3. Select any remark code from the “Results” table by clicking on it.
4. Click the “CO Add” button on the right side of the scene.



5. Repeat steps 3 and 4.
6. Click the “Copy” button a little bit above the “CO Add” button.



7. Select from the “Results” table one of the two remark codes which were copied together and understand that it has now gained one “association point” with the other remark code.

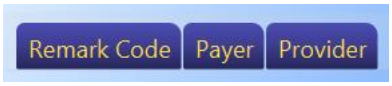
Typical Use

Searching for Information

The application starts by asking the user to agree to the licensing agreement. When the user has agreed, they are taken to the Information Search Menu.

For the user to use this menu to search for the desired information, the user should:

1. Navigate to the appropriate tab.



2. Click in the search text area for the tab

3. Type a search phrase.
4. Hit enter on the keyboard.

Making Suggestions

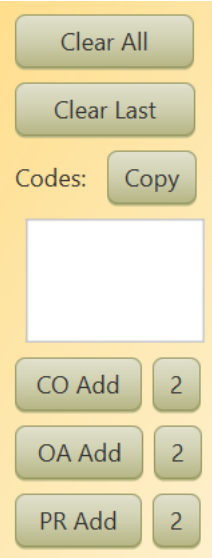
Each tab has a *Suggestions* button where the user may click in order to make any suggestions for the organization to view later.

Unique to Remark Code Tab

The remark code tab has a few unique features:

- The *T* column represents the total number of times that the remark code has been used with all stored search phrases. Importantly, because more than one search phrase can be associated with a remark code at once, it is possible – and likely – that this column will increment by a number greater than one when a *Copy* or *Add* button is used.
- The ability to copy remark codes to the clipboard, sometimes with the associated prefix by using the copy buttons in the below picture.
    - If the remark code's tag is CARC, the remark code will be copied with a prefix.
    - If the remark code's tag is RARC, the remark code will be copied without a prefix.

- The ability to add remark codes to a list view so that they may all be copied to the clipboard, separated by commas, sometimes with the associated prefix by using the add buttons in the below picture.
    - If the remark code's tag is CARC, the remark code will be added to the list with a prefix.
    - If the remark code's tag is RARC, the remark code will be added to the list without a prefix.
    - When the *Copy* button is clicked, all remark codes on the list will be copied to the user's clipboard, separated by commas.
    - The *Clear Last* button clears the remark code which was last added to the list
    - The *Clear All* button clears all remark codes on the list
    - If the user would like to add a remark code from the "Results" table, the user may use the *Add* buttons.
        - If, however, the user would like to add a remark code from the "Most Commonly Used With Your Selection" table, the user must use the *2* buttons next to the *Add* buttons with their desired prefix.



Unique to Payer Tab

The user may use the *Copy Name to Enter* button to copy the name to enter of the selected payer.

The user may also suggest a new payer combo to be added to the list of all payers; this will be seen by admins in the future and they may approve, edit and approve, or deny the suggestion.

Unique to Provider Tab

The user may use the *Copy NPI* button to copy the NPI the selected payer.

Administrator Functions

Admin Information Search

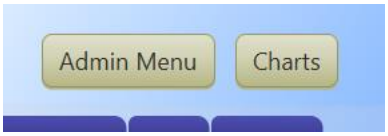
This application allows the user to log in as an administrator to access certain administrative features. When a user logs in as an administrator, the user is directed to the admin information search tab; this tab includes all the same features as the standard Information Search tab, as well as the following features:

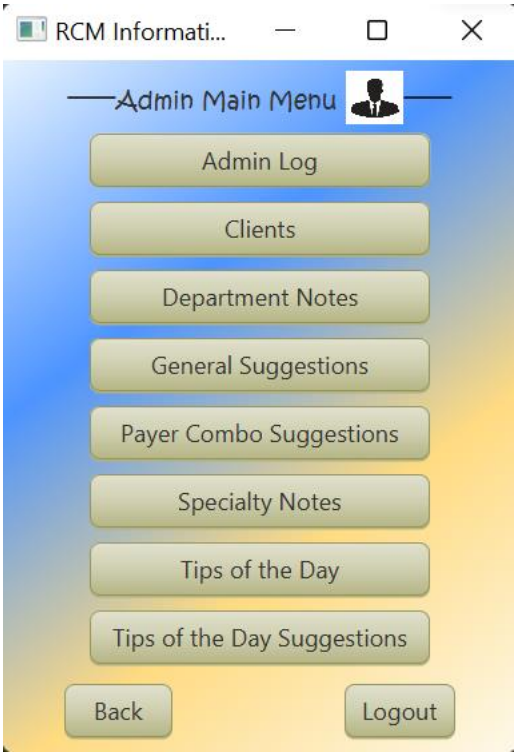
- The ability to add, edit, and remove remark codes, payers, and provides directly from their respective tabs.
- The ability to access the Admin Menu

To login, the user clicks the *Admin* button.

Admin Menu

The Admin Menu allows the user to view certain information vital to the business function of this application. To access the Admin Menu, the user clicks the *Admin Menu* button on which replaced the *Admin* login button upon login.





In the Admin Menu, the user can see all things listed in the above image, including all suggestions submitted by users, a log of all administrator actions, the tips of the day which are offered to users, clients, and departments, etc.

In addition to being able to view the associated items, when the administrator navigates to one of these windows by clicking on the associated button, they will be able to add, edit, or delete any of these items.

The administrator also can logout from this window.

Charts

The user may access the charts from the information search and admin information search tabs by clicking on the *Charts* button.

The charts show the remark codes that have been used (by copying them to the clipboard or adding them to a list to be copied later) the most times with any/all search phrases ever used. If ten remark codes have not been used in such a manner, the charts will show only the remark codes which have been used in such a way.

The available charts are:

- Bar Chart
- Pie Chart
- Scatter Chart

Code Summary – Why You Should Hire Me

This code is meant to get me a job and in it I put everything I could with respect to what employers want to see. Take a look at the JavaDocs to see the code – located in the same folder from the "Running the Code" section; here's a quick summary:

- Lots of hash maps to decrease search time and for ease of coding.
- Interfaces, abstract classes, parent-child relationships among classes, plenty of inheritance.
- Time zone conversions – everything stored in UTC time and converted to the time zone on the user's machine.
- There's code to test my code, so that when I make a change, I can run the tests to decrease the chance of a release with bugs.
- Tons of experience using the debugging tools and IDE in general.
- Timer class used to keep track of all searches within the last five minutes.
- Machine learning – specifically reinforced learning – which occurs in the background as the user naturally uses the application.
- Allows the user to search for data and presents that data in an organized, ordered, pleasing, and useful way to users.
- Allows users to manipulate the way the data is presented and organized.
- Code is, of course, commented and organized; the data, business, and interface layers are appropriately separated.
- Styled with CSS.
- JDBC used to connect with SQLite database, and the code is structured in such a way that its additions could easily be made to accommodate a different database vendor.
- Made with love, with only my development tools and my search engine.

Finally, I am currently in the process of turning this application into a website with Spring Boot!