

Spring 2024

Rhizomatica RAI Documentation

Gabriel Miki

Felix R. Millan

Front-end

The files below don't require very complex logic, they are mostly used for displaying things to the user:

- Platform ("platform" is not a file, it's a tab on the navigation bar)
 - Site.php
 - account.php
 - platform_stats.php
 - network_stats.php
- Statistics
 - cost_stats.php
 - call_stats.php
- Subscribers.php
 - Provisioning.php
- Resellers.php
 - provisioning.php
 - credit.php
 - credit_history.php
 - transactions.php
 - configuration.php
- Credit.php
 - add
 - transfer
 - history
 - status
- SMS.php

Spring 2024

- sendSMS.php
- CDR.php
- Rates.php

Most of these files are simple database queries or simple PHP scripts, that require the use of the files in the back-end section.

RAI Login Page GUI

Rhizomatica Administration Interface is composed of these four components the username field allows the user to input a string and the string in the password field is automatically hidden

- Most of the gui is found in login.php
- Username label with a textbox
- Password label with a textbox
- The language selection dropbox allows the user to select between three different languages and will change the language of the entire site
- Login button

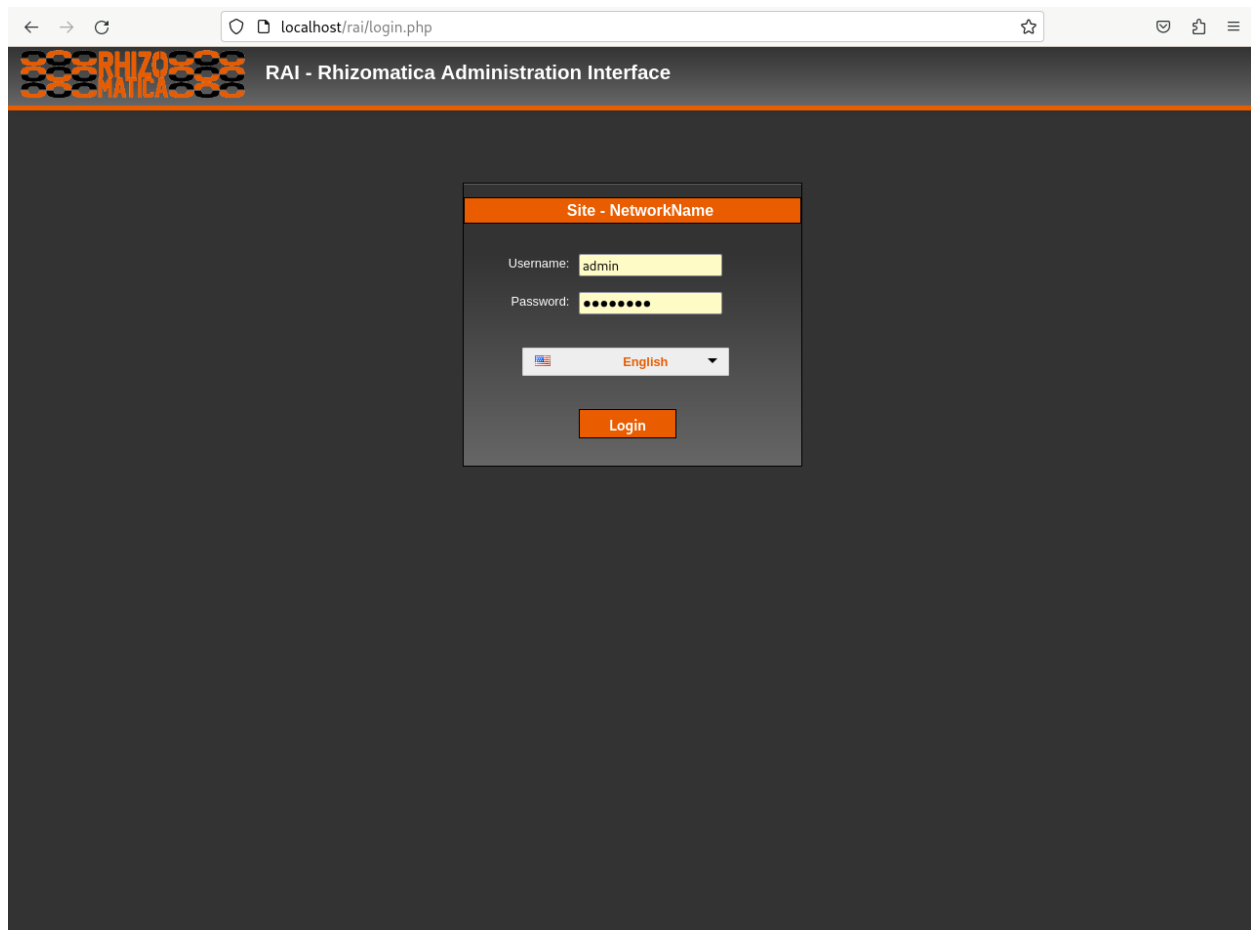


Figure 1.0

Rai Login page PHP

Login.php is mostly used as the UI that implements other functions from other classes that handle more important things, such as the login logic and the page redirect logic. In total, this file requires three additional files, but the two most important ones are **session.php** and **configuration.php**. While this file does contain some POST checking, it's still very simple, and

does not handle input sanitization or validation, however, this is handled in the other files. If we were to build this from scratch, this is a general idea on how it could be accomplished:

- Create input field for **username** and **password**
- Create drop-down menu for the **language** selection
- Create login button that does the following:
 - Validates **username** and **password**
 - Sets the language to the one specified (Figure 1.1)
 - If everything goes well, redirect to the main page, otherwise, display an error message, and stay on login page (Figure 1.2)

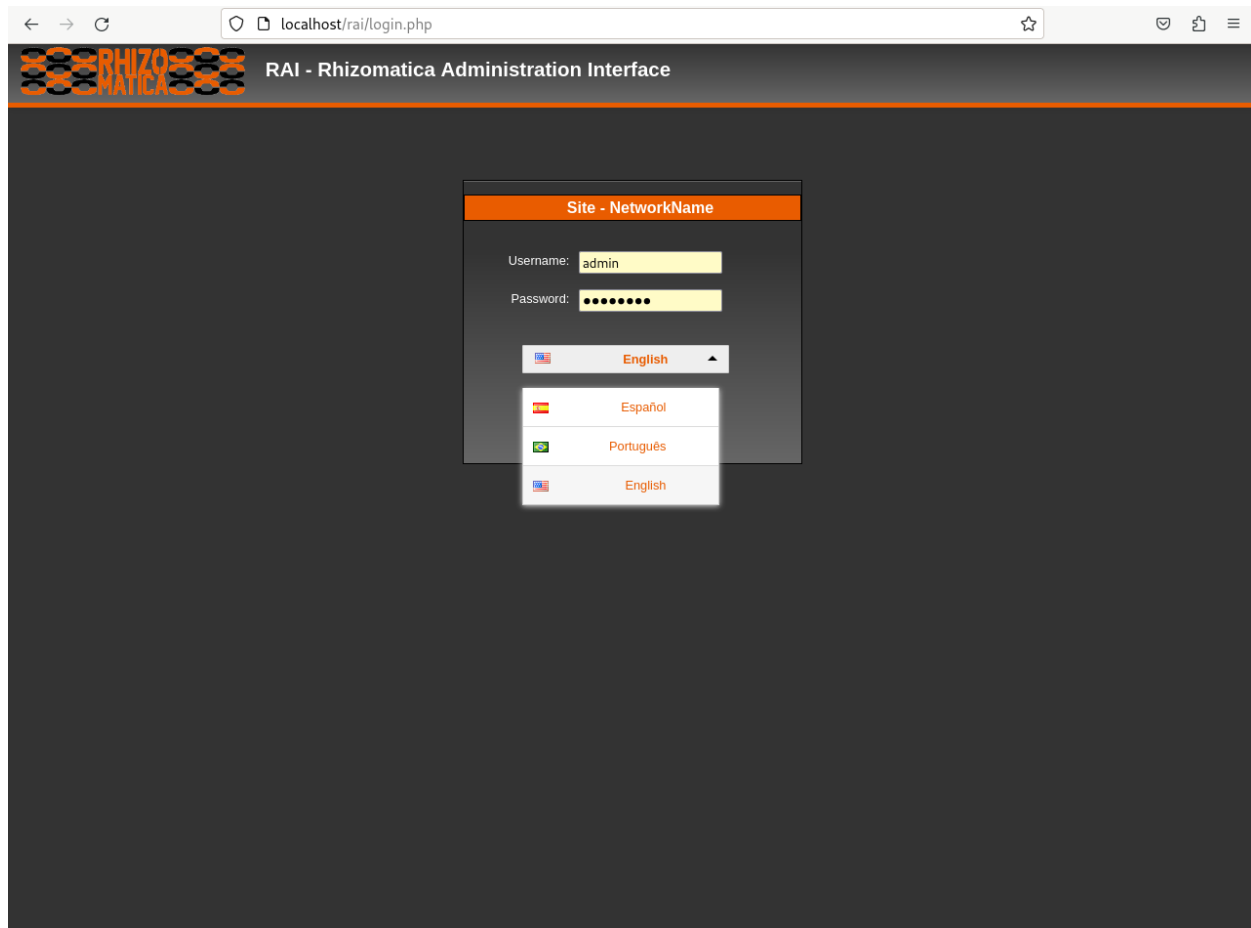


Figure 1.1

← → ↻ localhost/rai/login.php ☆

RAI - Rhizomatica Administration Interface

Site - NetworkName

Login incorrect

Username: admin

Password: ••••••••

English

Login

Figure 1.2

Rhizomatic Navbar

The main menu has a navbar containing all of the following elements as shown in (figure 2.0)

When signed in the Subscribers will be automatically selected and displayed

- Clicking on the top left image takes you to a blank page as shown in (figure 2.1)
- Logout when clicked takes the user back to the Sign Screen as shown in (figure 1.1)
- Platform when clicked does nothing on hover and displays the following options (figure 3.0)
 - Site (figure 3.1)
 - Voip Account (figure 3.2)
 - System Stats (figure 3.3)
 - Network Stats (figure 3.4)
 - Online Support (figure 3.5)
- Statistics when clicked refreshes the current page on hover and displays the following options
 - Casts (figure 4.0)
 - Calls (figure 4.1)
- Subscribers when clicked takes you to the subscribers page (figure 5.0) on hover displays the following option
 - Provisioning (figure 5.1)
- Resellers when clicked takes you to the resellers page (figure 6.0) on hover displays the following options
 - List resellers (figure 6.1)
 - Provisioning (figure 6.2)
 - Add Credit (figure 6.3)
 - Reseller invoices (figure 6.4)
 - Transaction (figure 6.5)
 - Configuration (figure 6.6)
- Credits when clicked refreshes the current page on hover and displays the following options
 - Add Credit (figure 7.0)
 - Transfer Credit (figure 7.1)
 - History (figure 7.2)
 - Status (figure 7.3)
- SMSs when clicked takes you to the SMSs page (figure 8.0) on hover displays the following option
 - Send SMS (figure 8.1)
- CDRs when clicked take you to the SMSs page (figure 9.0)
- Rates when clicked takes you to the Rates page (figure 10.0)



Figure 2.0

Spring 2024

Rhizomatica-Index

- This is the blank page that the user is taken to when the user clicks on the Rhizomatica image on the top right (Figure 2.1)

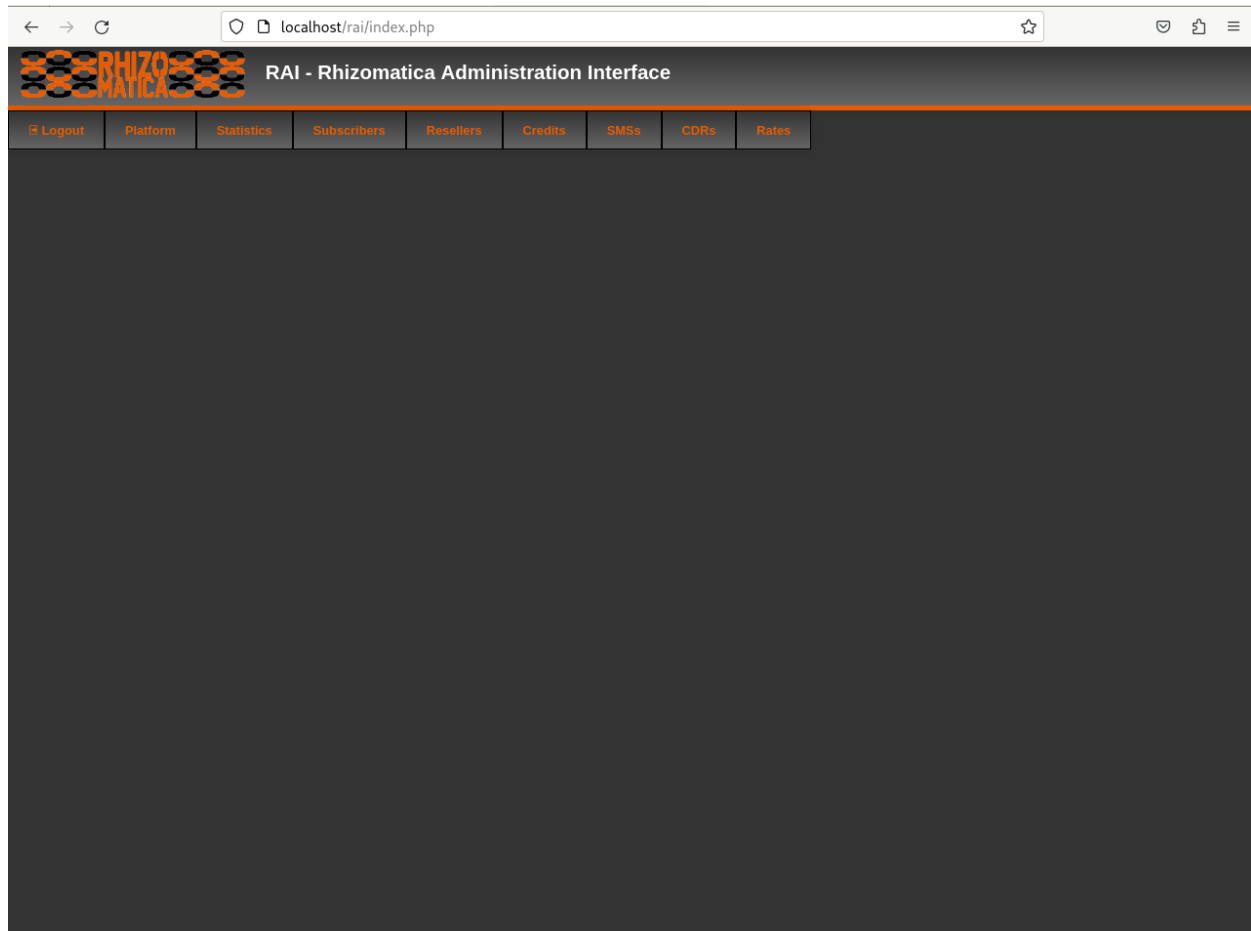


Figure 2.1

Spring 2024

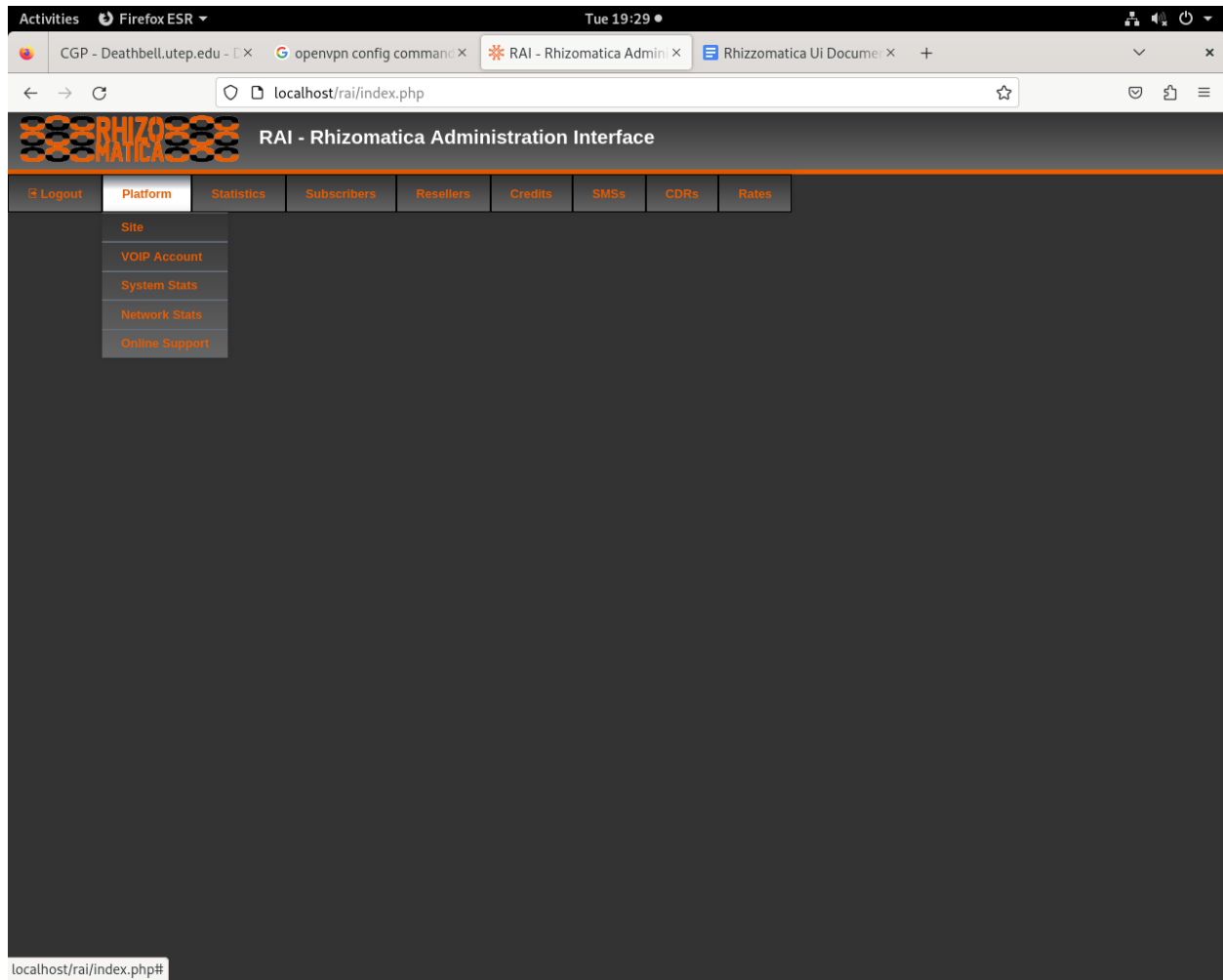


Figure 3.0

Spring 2024

Platform-Site

- This page is fully functional displaying all the correct values for the site (Figure 3.1).
- This gui element is contained in site.php
- At the moment, they appear to be placeholders

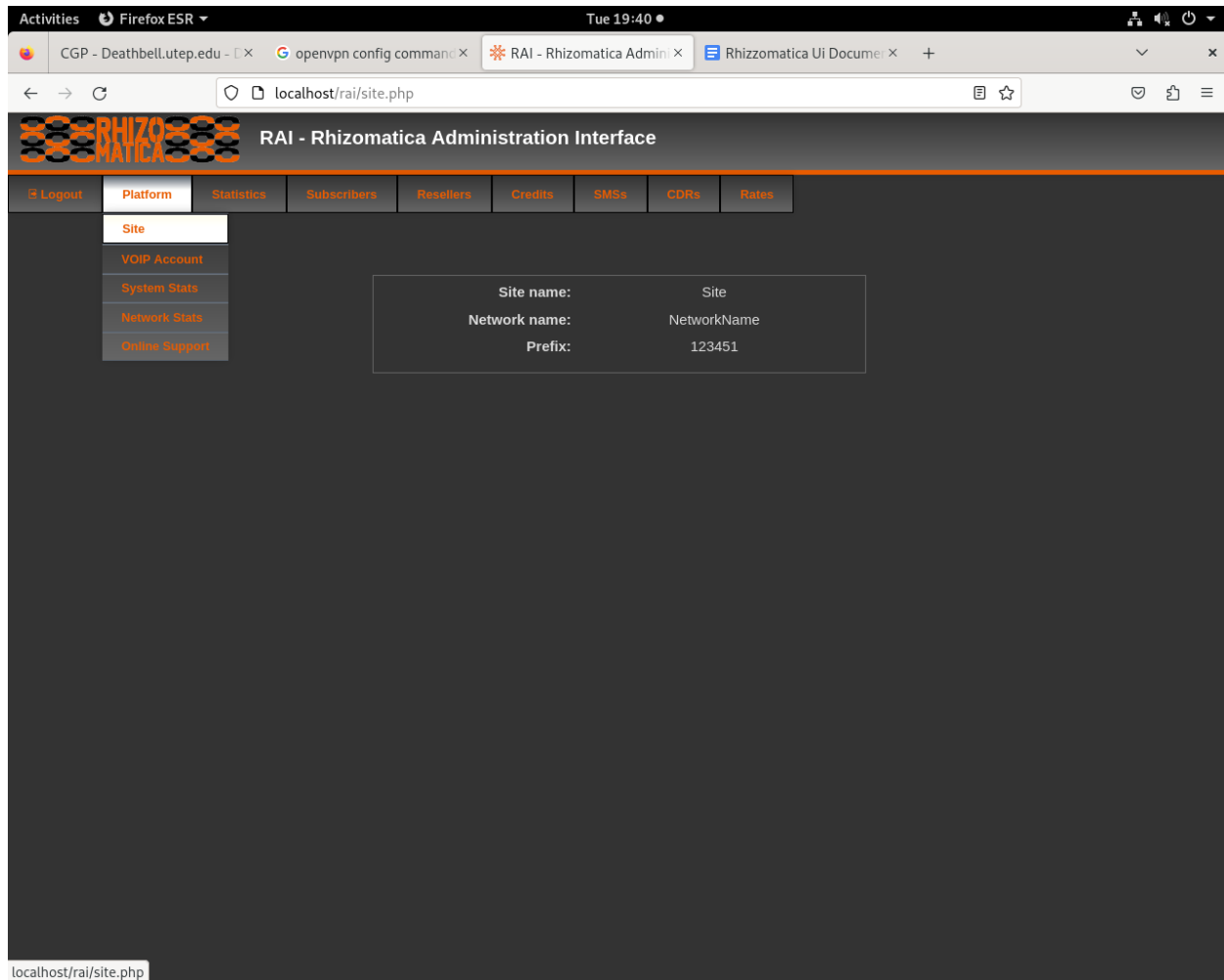


Figure 3.1

Platform - Voip Account

- This page is contained in account.php
- This page seems to be for displaying a Rhizomatic account and how many credits it has allocated to it.
- This page is properly displaying however all of the values under credit are the default values cause it has no information to query (Figure 3.2).

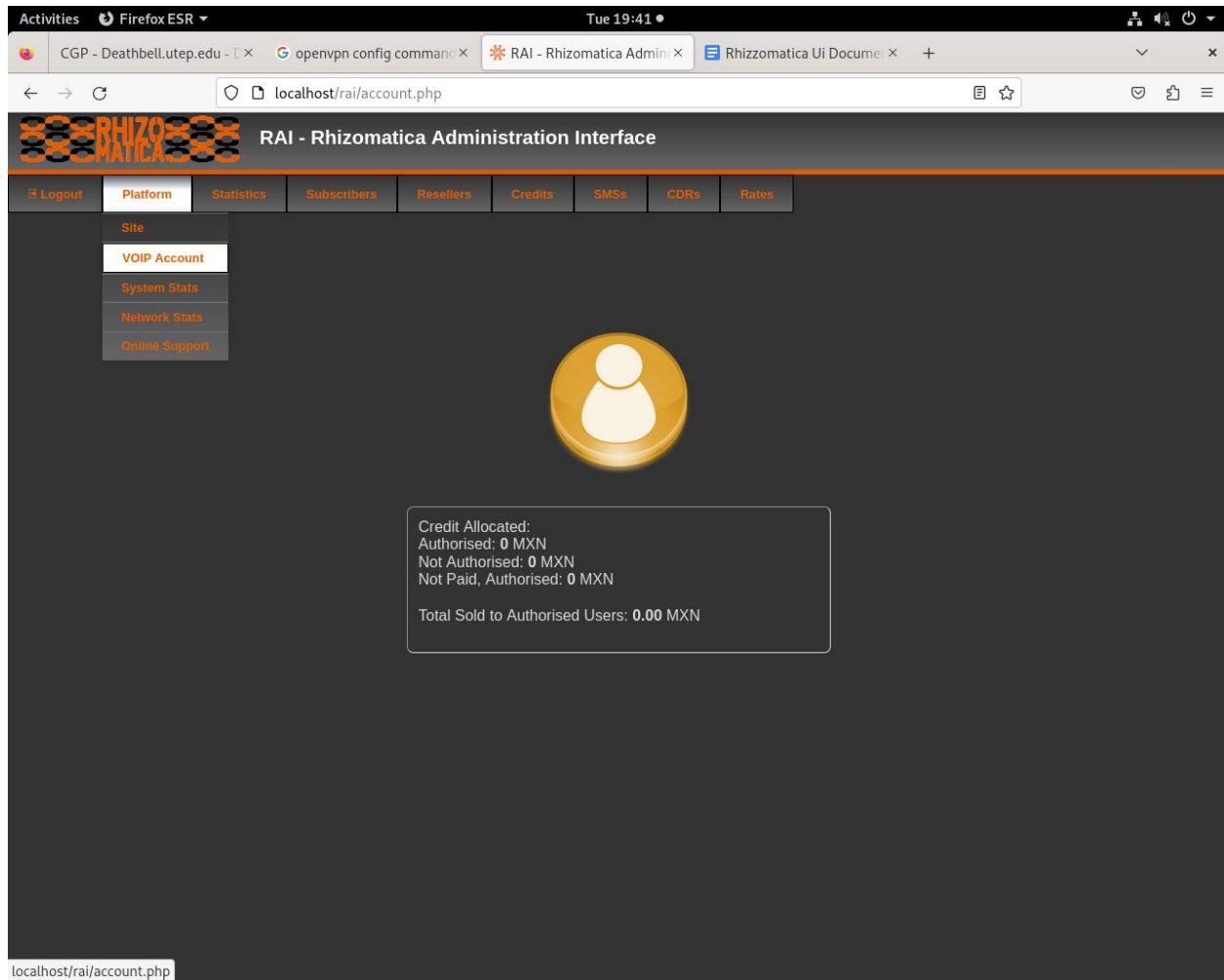


Figure 3.2

Platform-System Stats

- This page is contained in platform_stats.php
- This page is functional and contains nine graphs each displaying their respective information (Figure 3.3).
- It also contains 4 buttons Last 12 Hours, Daily, Weekly, and Monthly all of these are functional and change each graph's x-axis to their respective values (Figure 3.3).
- Displays load, CPU, and memory usage. Interestingly, the graphs appear to be hardcoded PNG images that are simply retrieved with a simple php function.

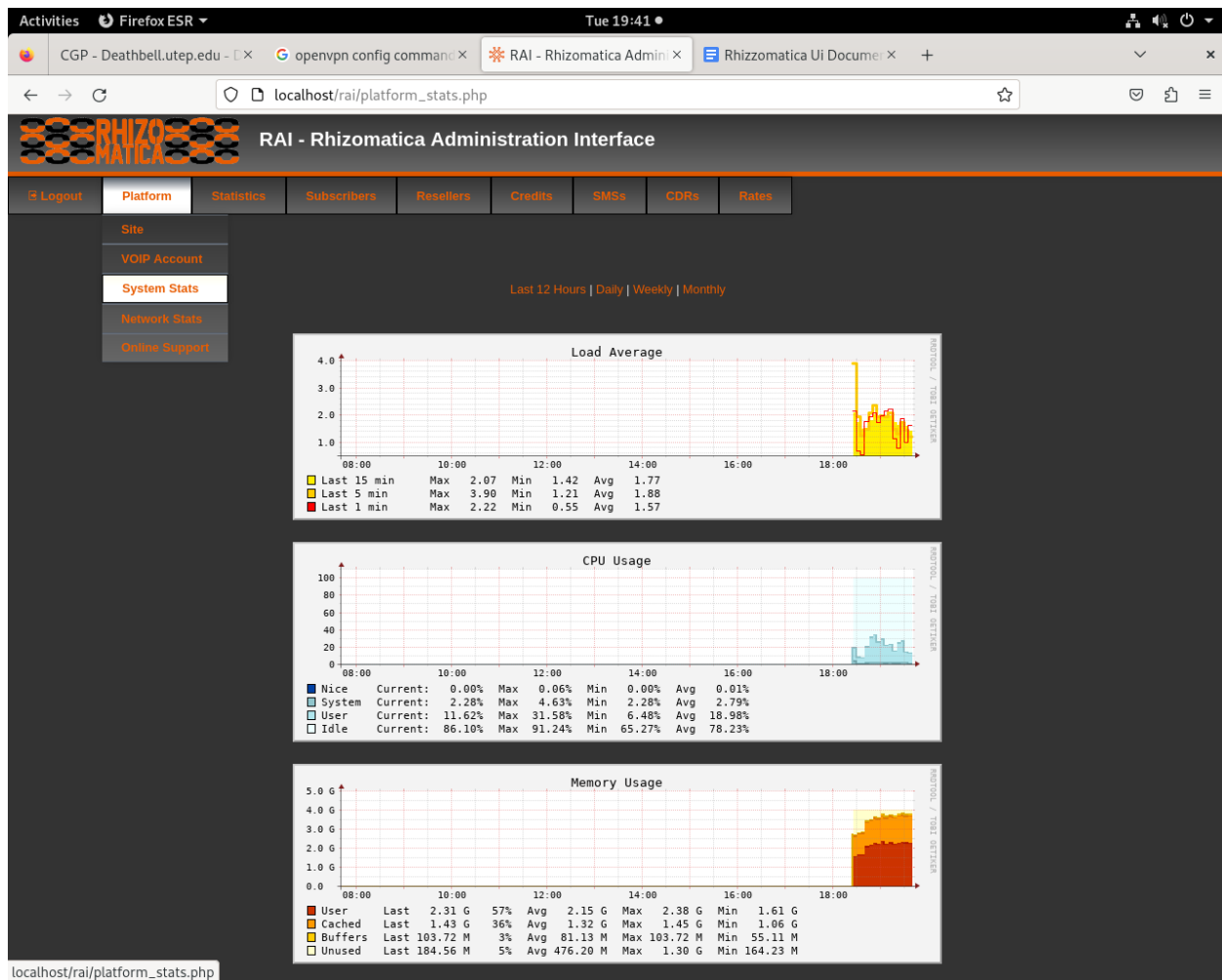


Figure 3.3

Platform-Network Stats

- This page is contained in network_stats.php
- This submenu contains thirteen graphs eleven properly displaying and two showing up only as a document icon; these display the network activity that has happened using the system. (figure 3.4)
- It contains 6 options Last 3 Hours, Last 12 Hours, Daily, Weekly, Monthly, Year, and Meas all these buttons do is change the x-axis of the graphs to their respective time frames with the exception of meas which takes you to a different page (figure 3.4.1)
- Similarly to platform_stats.php, this file displays the network statistics shown below

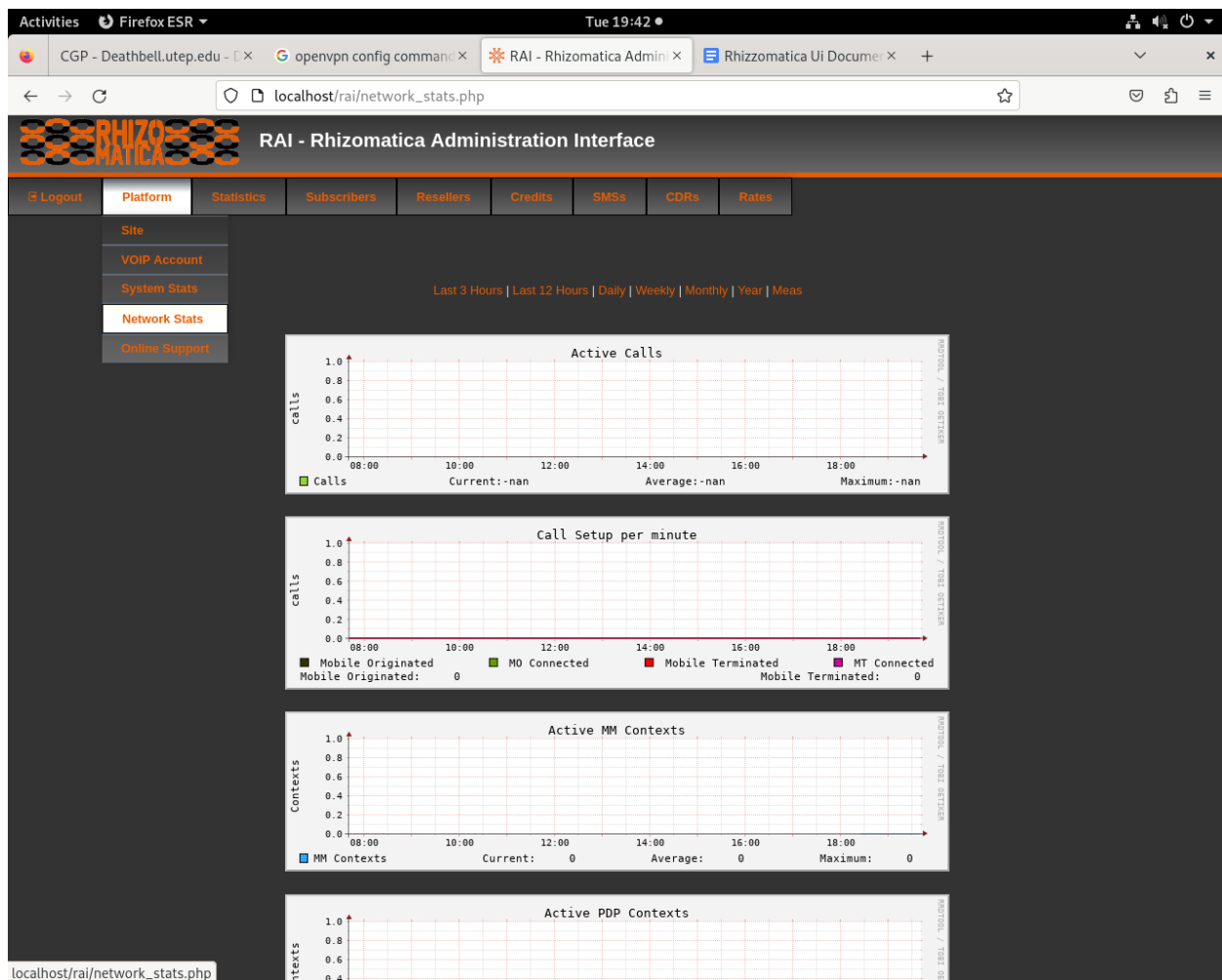


Figure 3.4

Spring 2024

Platform-Network Stats-Meas

- This page is nonfunctional as there doesn't seem to be anything registered within it and changing the drop-down or bottom buttons does not seem to change anything (Figure 3.4.1).
- Looks like network_stats? meas=1 is supposed to display measurements related to the signal, however it looks like it has not been implemented yet, and nothing seems to work at the moment.

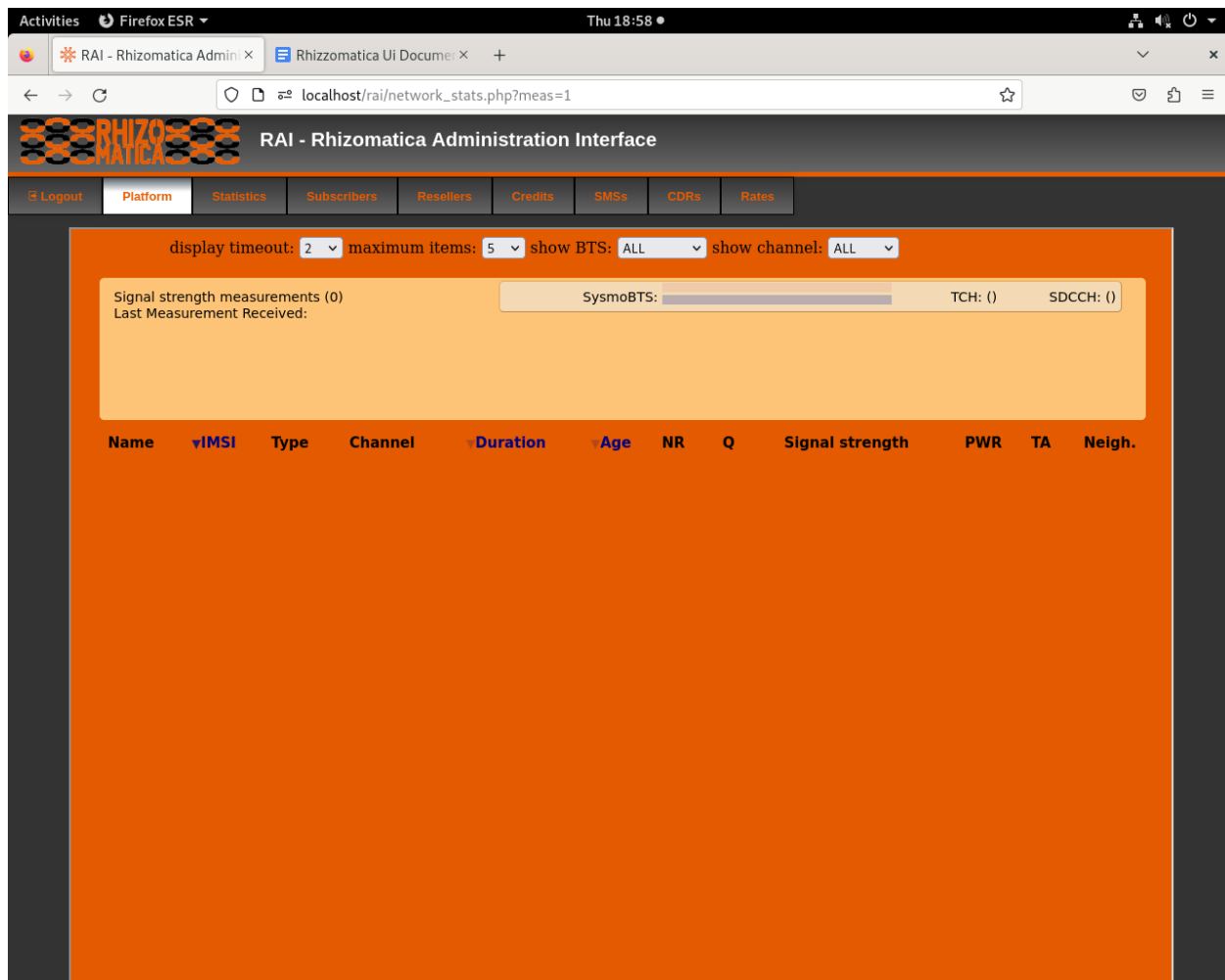


Figure 3.4.1

Platform- Online Support

- Clicking this option opens up a whole new tab allowing you to connect to a support hotline this seems to be separate from the Rhizomatica system (Figure 3.5).
- Takes user to a new tab where they can request support. Uses Kiwi IRC to function. We are unable to test the implementation at the moment, but based on their official website, it seems to support multiple channels and individual 1-on-1 chat sessions.

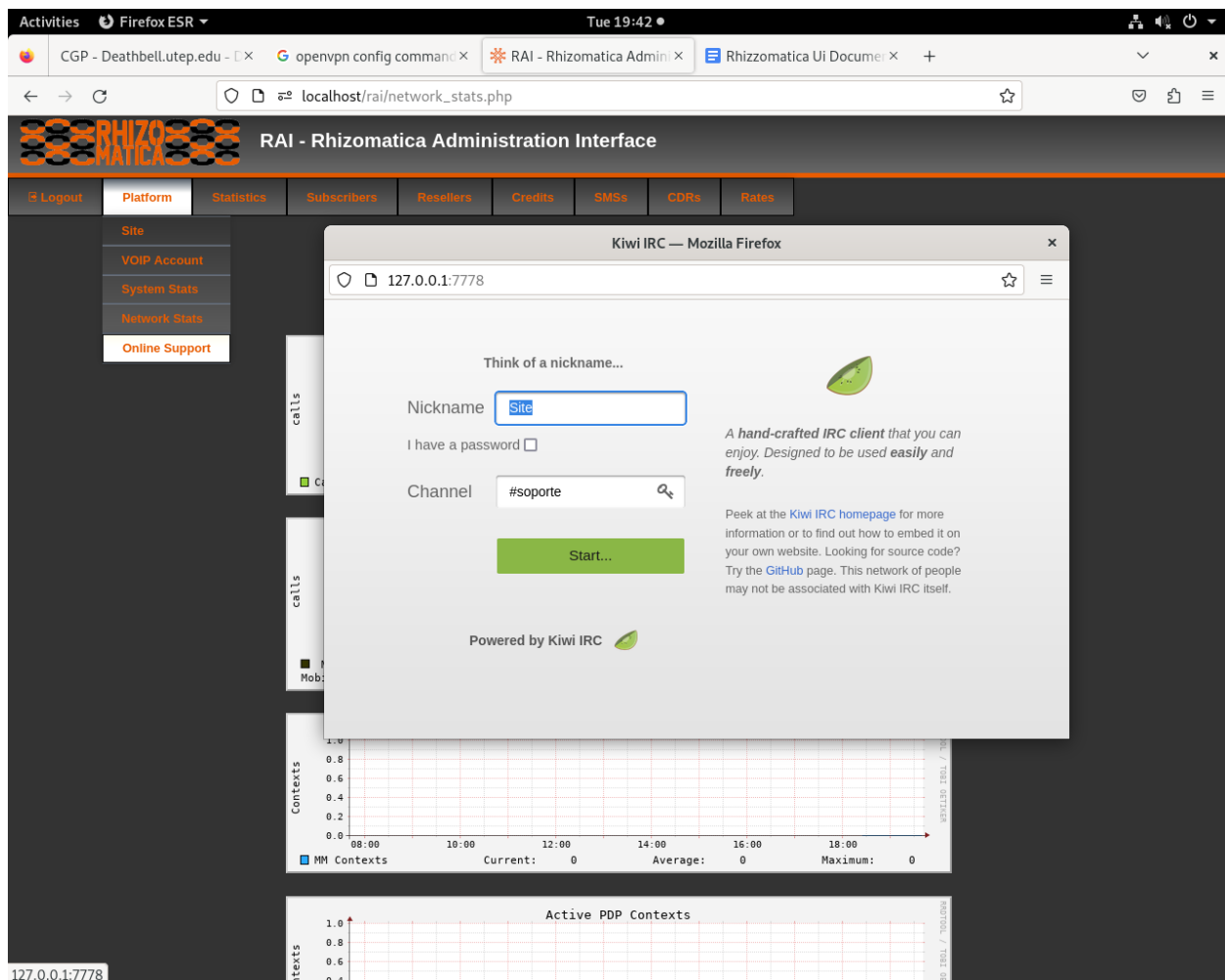


Figure 3.5

Statistics (tab)

The sub-links found on this tab seem to display the average of costs and calls from all users. They seem to be working relatively fine, however, there is nothing to display yet, which is why they look a little strange. Further testing needs to be done before determining whether it's fully functional or not.

Statistic-Cost

- This page is contained in cost_stats.php
- Seems to be mostly non-functional as the textbox it is displaying seems incomplete and there is no real indicator of what f: a and N: o mean as well as it seemingly fails to load two images at the bottom. (figure 4.0)
- It also contains three buttons at the very bottom labeled Last 7 Days, Last 4 Weeks, and Monthly respectively all of these buttons do nothing. (figure 4.0)
- cost_stats.php Its purpose is to display the cost average from all users. Interestingly, there are two pictures that failed to load at the bottom. The logic for displaying the pictures is the same logic as in the previous images that did work, so it is interesting to see that these failed to load (probably caused by the fact that there is nothing to display yet). It also allows the admin to filter by "last 7 days", "last 4 weeks", and "monthly".

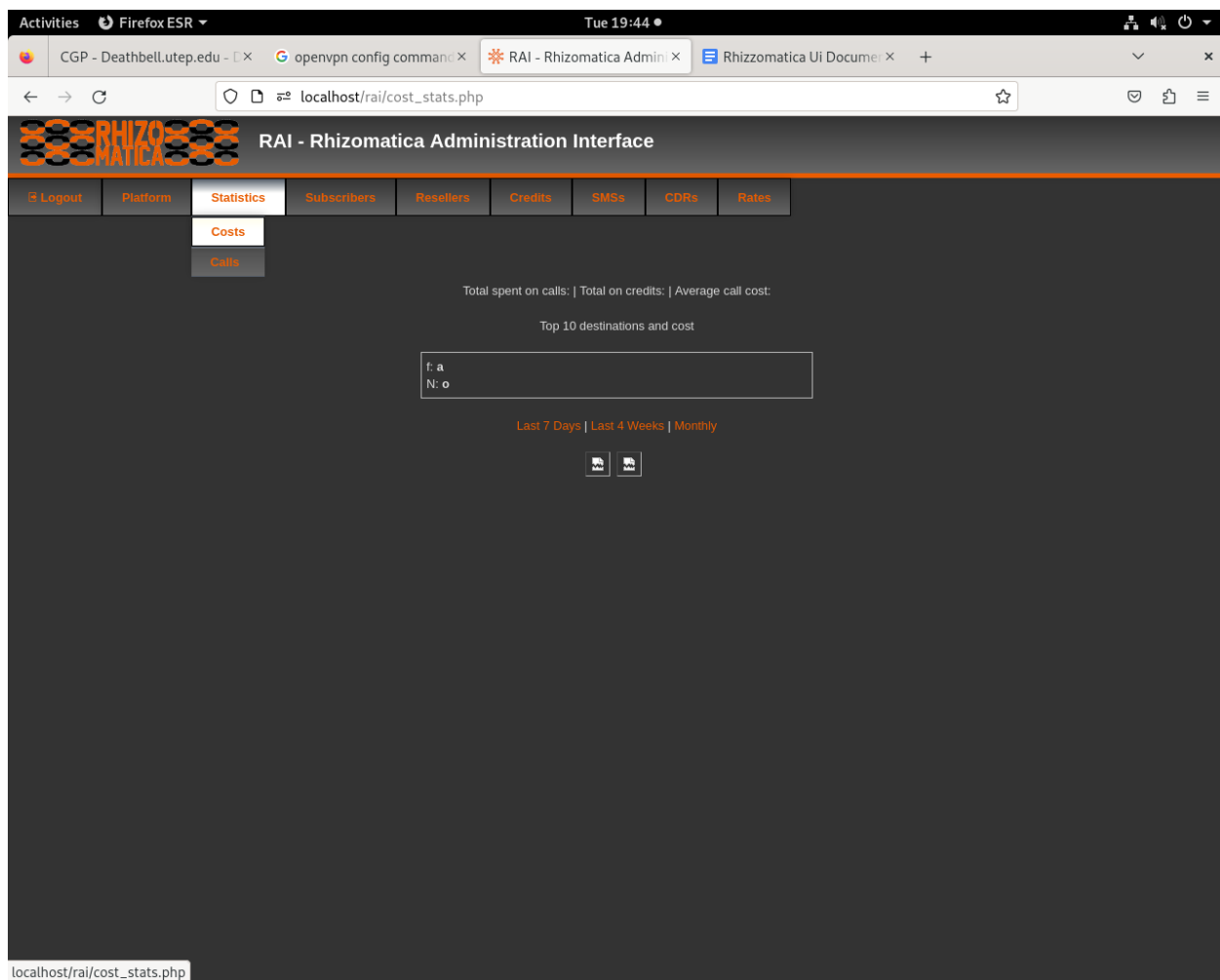


Figure 4.0

Statistics-Calls

- This page seems to be partially functional with their values associated with total calls and average call duration with total minutes being the only one missing.
- It has three buttons at the bottom labeled Last 7 Days, Last 4 Weeks, and Monthly by default the last 7 days option is selected which fails to load two graphs (figure 4.1). However Last 4 Weeks and Monthly both properly display graphs (figure 4.2 and Figure 4.2.1).
- these stats (figure 4.2 and Figure 4.2.1). are somewhat functional, and it's probable that the ones on cost_stats.php are also partially functional. It's definitely strange that the images are PNG files that already exist rather than somehow generating them, but it's probably a lot easier on the computation.

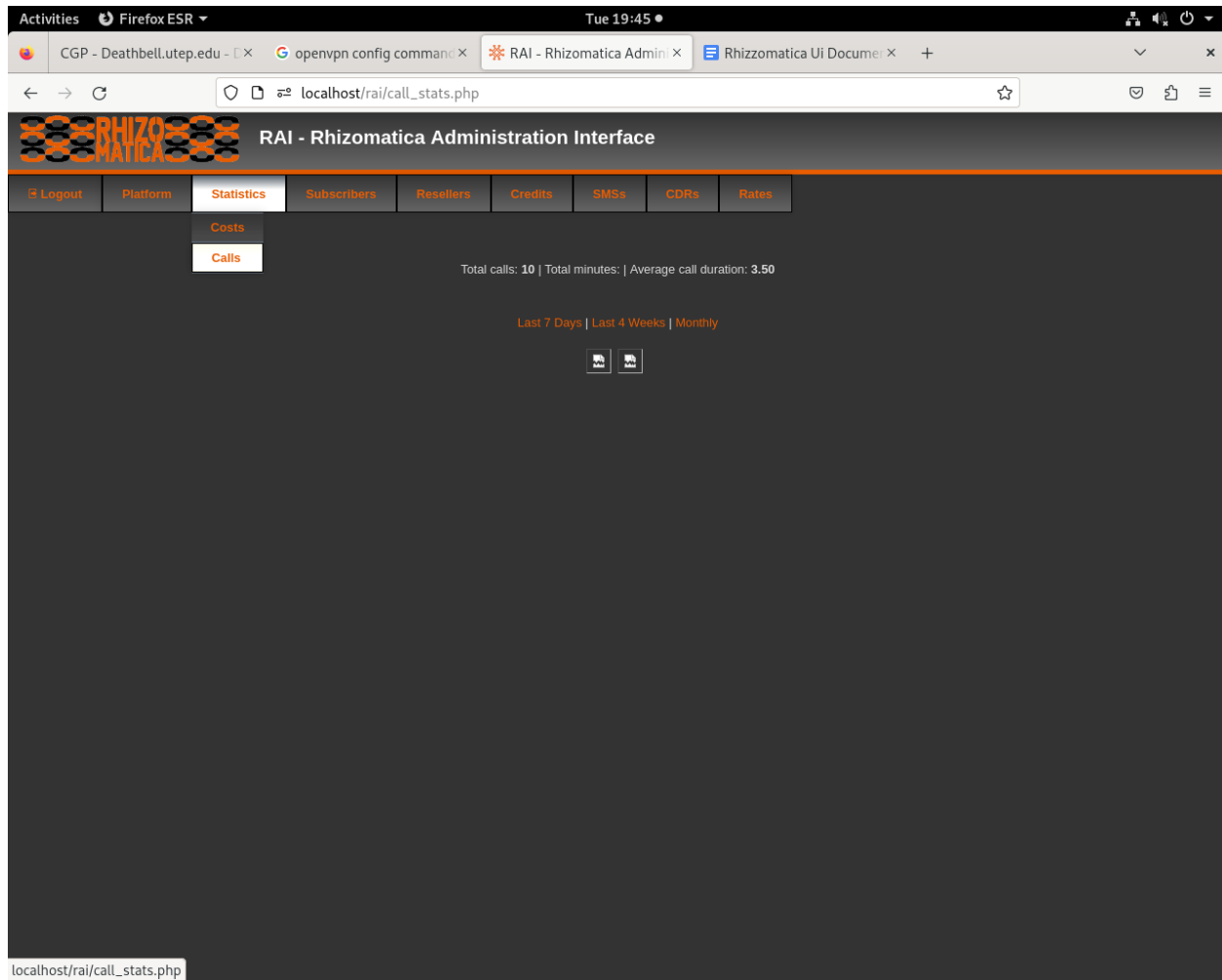


Figure 4.1

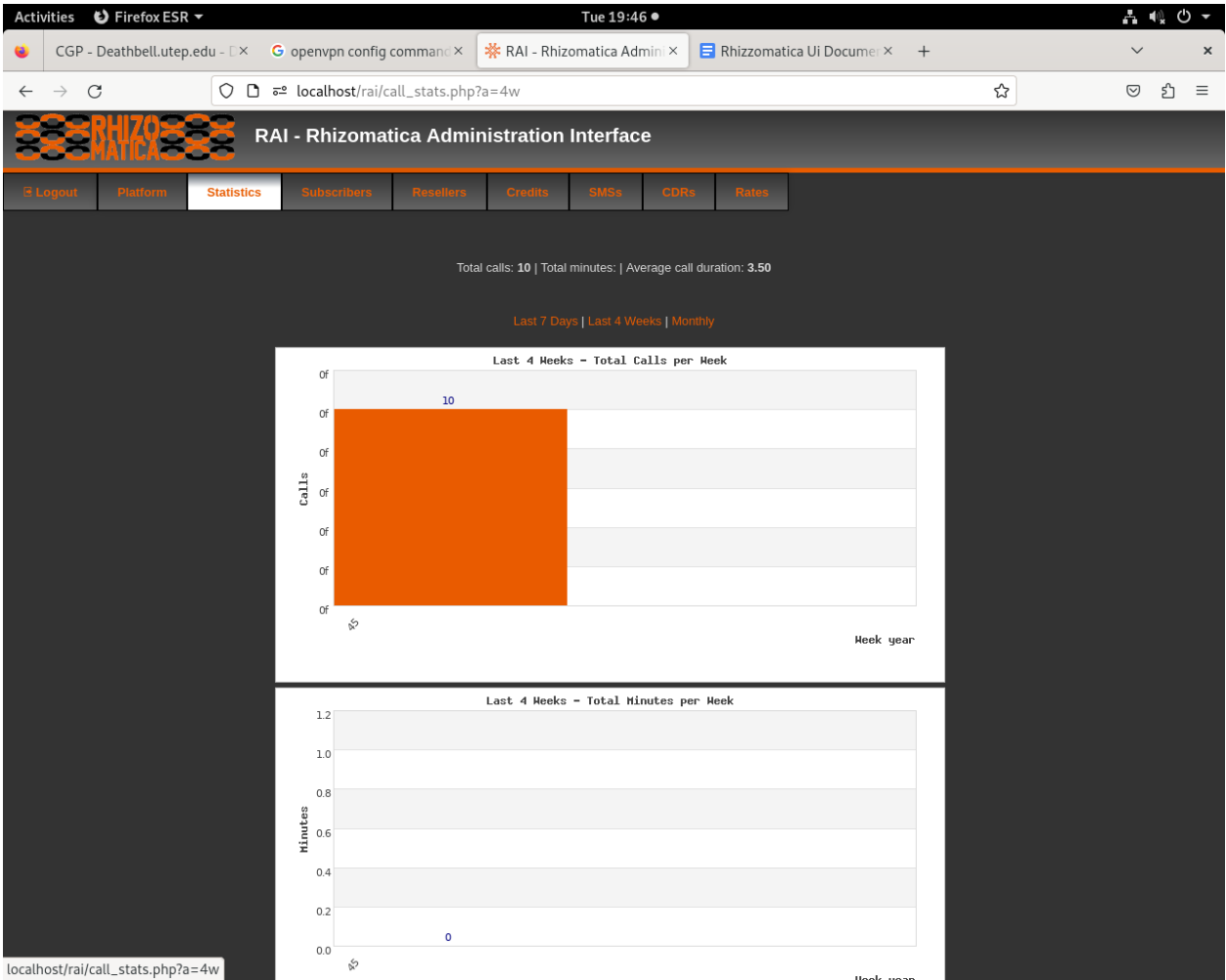


Figure 4.2

Spring 2024

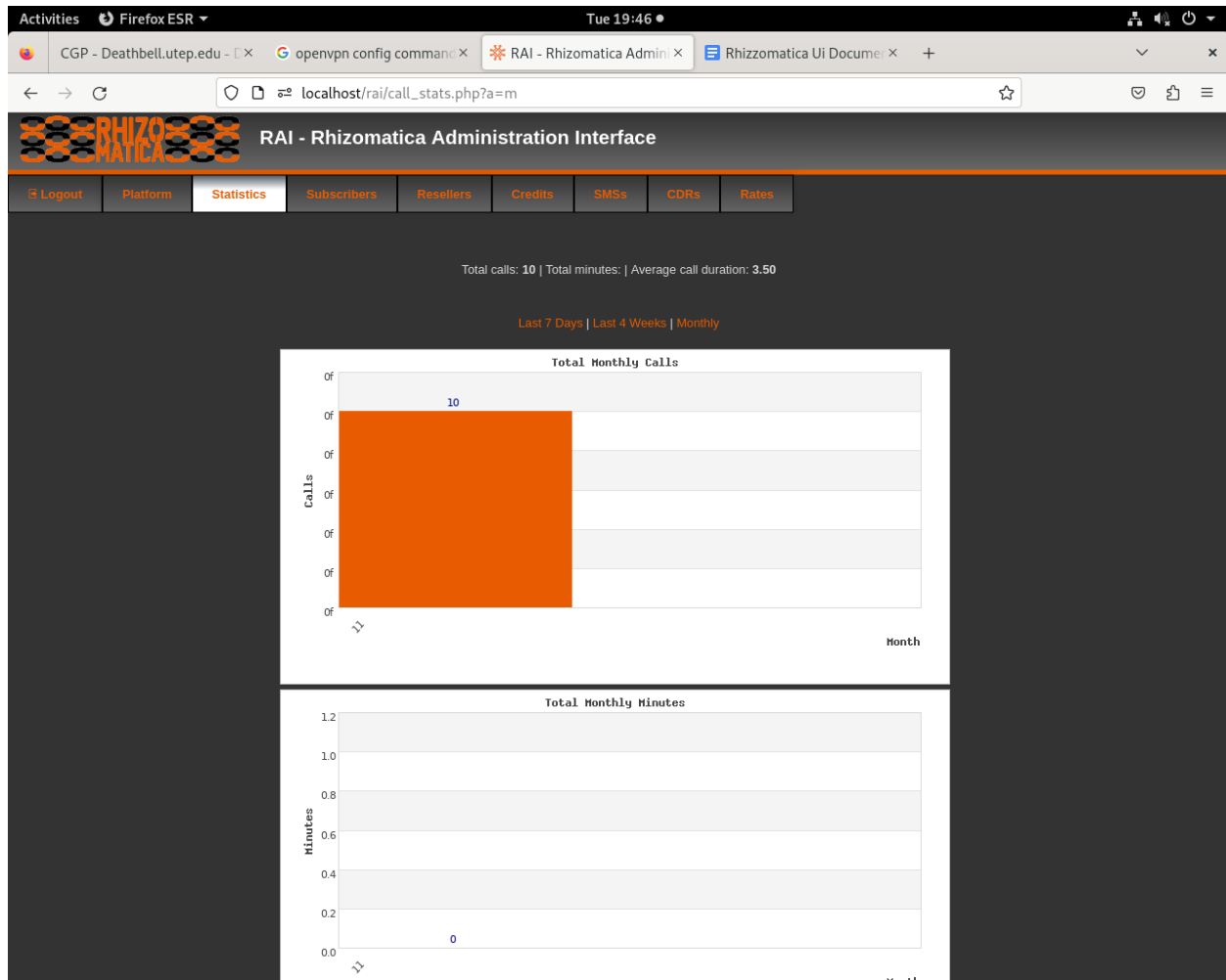


Figure 4.2.1

Subscribers

- This page is contained in Subscribers.php
- Displays to the user information about subscribers to the Rhizomatica system allows for different ways for the user to search and filter this data.
- The user can select how many total entries to show with the options being 10,25,50 and all this currently does nothing as there are only two entries (Figure 5.0.1).
- The user can also click on the arrows next to Activation date, Subscription date, Subscription Status, Authorized, Number, Name, and Balance. Location and Actions and flip the ordering of the list based on that one attribute. For example, clicking the arrow on a name will make it sorted in reverse alphabetical order. Only one arrow can be clicked at a time.
- The options Authorized and Number have drop downs with authorized having two images of locks allowing you to filter by their respective color locks (Figure 5.0.2). The number has 3 options: a green, yellow, and red button I don't know what this does as red is the only option that displays anything, and it's not clear what it's filtering by (Figure 5.0.3).
- The case does not matter when filtering using the search bar
- The search bar at the top right allows the user to search the table however it has several problems typing in dates or times will result in no data found (Figure 5.0.4).
- It will also display no result if you look up NOT_PAID or PAID two items both on the table
- Search will behave strangely if you type in 0 it will only show one record even though both contain zero (Figure 5.0.5).
- The search bar functions as expected when you search for an entry in the Number or name field.
- The clipboard option under actions opens a new screen that allows you to edit the entry.
- The trash can deletes the entry.
- The five buttons on the bottom left don't do anything as there are not enough entries in the table (Figure 5.0.6).

RAI - Rhizomatica Administration Interface

Logout Platform Statistics Subscribers Results Credits SMSs CDRs Rates

Unpaid subscription: 1 Unauthorized: 1 Paid subscription: 0 Credit Sold: \$0 Credit Used: \$0 Online: 0 (Roaming: 0) Offline: 1

Subscribers Phones

Show 10 entries

Activation date	Subscription date	Subscription Status	Authorized	Number	Name	Balance	Location	Actions
09-11-2023	01-01-2024 08:00:02	NOT_PAID		1234515555	test	0		

Showing 1 to 1 of 1 entries

First Previous Next Last

Spring 2024

Figure 5.0

Subscribers Phones

Show: **All** entries

Search:

Activation date	Subscription date	Subscription Status	Authorized	Number	Name	Balance	Location	Actions
09-11-2023	01-01-2024 08:00:02	NOT_PAID		12345155555	test	0		
20-02-2024	20-02-2024 19:47:14	PAID		123451sssss	dssss	0		

Figure 5.0.1

RAI - Rhizomatica Administration Interface

Logout Platform Statistics **Subscribers** Resellers Credits SMSs CDRs Rates

Unpaid subscription: 1 Unauthorized: 1 Paid subscription: 1 Credit Sold: \$0 Credit Used: \$0 Online: 0 (Roaming: 0) Offline: 2

Subscribers Phones

Show: **All** entries

Search:

Activation date	Subscription date	Subscription Status	Authorized	Number	Name	Balance	Location	Actions
09-11-2023	01-01-2024 08:00:02	NOT_PAID		12345155555	test	0		
20-02-2024	20-02-2024 19:47:14	PAID		123451sssss	dssss	0		

Showing 1 to 2 of 2 entries

First Previous **1** Next Last

Figure 5.0.2

RAI - Rhizomatica Administration Interface

Logout Platform Statistics **Subscribers** Resellers Credits SMSs CDRs Rates

Unpaid subscription: 1 Unauthorized: 1 Paid subscription: 1 Credit Sold: \$0 Credit Used: \$0 Online: 0 (Roaming: 0) Offline: 2

Subscribers Phones

Show: **All** entries

Search:

Activation date	Subscription date	Subscription Status	Authorized	Number	Name	Balance	Location	Actions
09-11-2023	01-01-2024 08:00:02	NOT_PAID		55555	test	0		
20-02-2024	20-02-2024 19:47:14	PAID		sssss	dssss	0		

Showing 1 to 2 of 2 entries

First Previous **1** Next Last

Figure 5.0.3

Spring 2024

Activities Firefox ESR Thu 20:23

RAI - Rhizomatica Administration Interface

Logout Platform Statistics **Subscribers** Resellers Credits SMSs CDRs Rates

Unpaid subscription: 1 Unauthorized: 1 Paid subscription: 1 Credit Sold: \$0 Credit Used: \$0 Online: 0 (Roaming: 0) Offline: 2

Subscribers Phones

Show 10 entries Search: 20-02-2024

Activation date	Subscription date	Subscription Status	Authorized	Number	Name	Balance	Location	Actions
No matching records found								
Activation date	Subscription date	Subscription Status	Authorized	Number	Name	Balance	Location	Actions

Showing 0 to 0 of 0 entries (filtered from 2 total entries)

First Previous Next Last

Figure 5.0.4

localhost/rai/subscribers.php

RAI - Rhizomatica Administration Interface

Logout Platform Statistics **Subscribers** Resellers Credits SMSs CDRs Rates

Unpaid subscription: 1 Unauthorized: 1 Paid subscription: 1 Credit Sold: \$0 Credit Used: \$0 Online: 0 (Roaming: 0) Offline: 2

Subscribers Phones

Show 10 entries Search: 0

Activation date	Subscription date	Subscription Status	Authorized	Number	Name	Balance	Location	Actions
09-11-2023	01-01-2024 08:00:02	NOT_PAID		12345155555	test	0		
Activation date	Subscription date	Subscription Status	Authorized	Number	Name	Balance	Location	Actions

Showing 1 to 1 of 1 entries (filtered from 2 total entries)

First Previous Next Last

Figure 5.0.5

Spring 2024

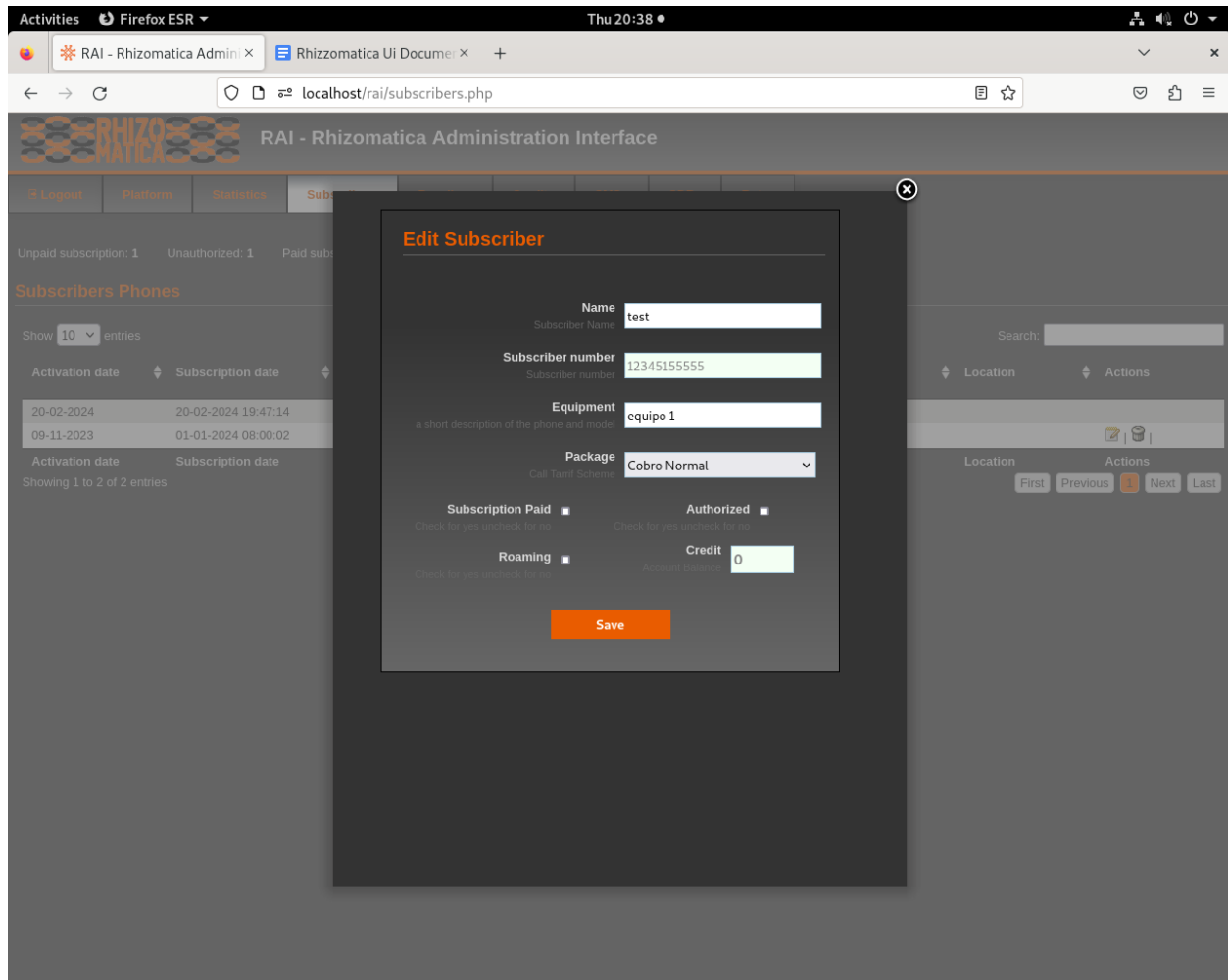


Figure 5.0.6

Subscribers - Provisioning

- This page allows the user to add new subscribers using two formats if the subscriber has a subscriber number or an IMSI(Figure 5.1).
- If they do not have a subscriber number or an IMSI they are added using an IMEI (Figure 5.1.1)

The screenshot displays a web browser window at the URL `localhost/rai/provisioning.php`. The page title is "RAI - Rhizomatica Administration Interface". The navigation bar includes links for Logout, Platform, Statistics, Subscribers, Resellers, Credits, SMSs, CDRs, and Rates. The "Subscribers" tab is active, and a "Provisioning" sub-tab is visible. The main content area features a form titled "Provision a new subscriber". The form has two tabs: "IMSI" (selected) and "IMEI". Below the tabs, there is an error message: "Error getting IMSI please retry". The form contains three input fields: "Name" (with the label "Subscriber Name" below it), "Subscriber number or IMSI" (with the label "Subscriber number" below it), and "Equipment" (with the label "a short description of the phone and model - iPhone5/Galaxy 7/etc" below it). An "Add" button is located at the bottom right of the form.

Figure 5.1

Spring 2024

The screenshot shows a web browser window with the address bar displaying `localhost/rai/provisioning.php`. The page title is "RAI - Rhizomatica Administration Interface". The navigation bar includes links for Logout, Platform, Statistics, Subscribers (active), Resellers, Credits, SMSs, CDRs, and Rates. The main content area features a form titled "Provision a new subscriber". The form has a header with "IMSI" and "IMEI" tabs. Below the tabs, there are three input fields: "Name" (with a sub-label "Subscriber Name"), "IMEI (*#06#)", and "Equipment" (with a sub-label "a short description of the phone and model - iPhone5/Galaxy 7/etc."). An "Add" button is located at the bottom right of the form. A note states: "The phone must be connected or have recently attempted to connect".

RAI - Rhizomatica Administration Interface

Logout Platform Statistics Subscribers Resellers Credits SMSs CDRs Rates

Provision a new subscriber

IMSI IMEI

Name
Subscriber Name

IMEI (*#06#)

The phone must be connected or have recently attempted to connect

Equipment
a short description of the phone and model - iPhone5/Galaxy 7/etc.

Add

Figure 5.1.1

Spring 2024

Activities Firefox ESR Thu 18:37

RAI - Rhizomatica Administration Interface

Logout Platform Statistics Subscribers Resellers Credits SMSs CDRs Rates

Resellers

Show 10 entries

Creation Date Reseller Number

Search:

Creation Date Reseller Number PIN Balance Total Sales Actions

No data available in table

Showing 0 to 0 of 0 entries

First Previous Next Last

localhost/rai/resellers.php

Figure 6.0

Spring 2024

Resellers

- Its purpose seems to be to display resellers, however, there are none to be displayed (it's not even clear what a reseller refers to in this context).
- The reseller displays an empty table to the user as there seems to be no data to display so it cannot be tested. It appears to have similar options to subscribers and probably shares similar issues.

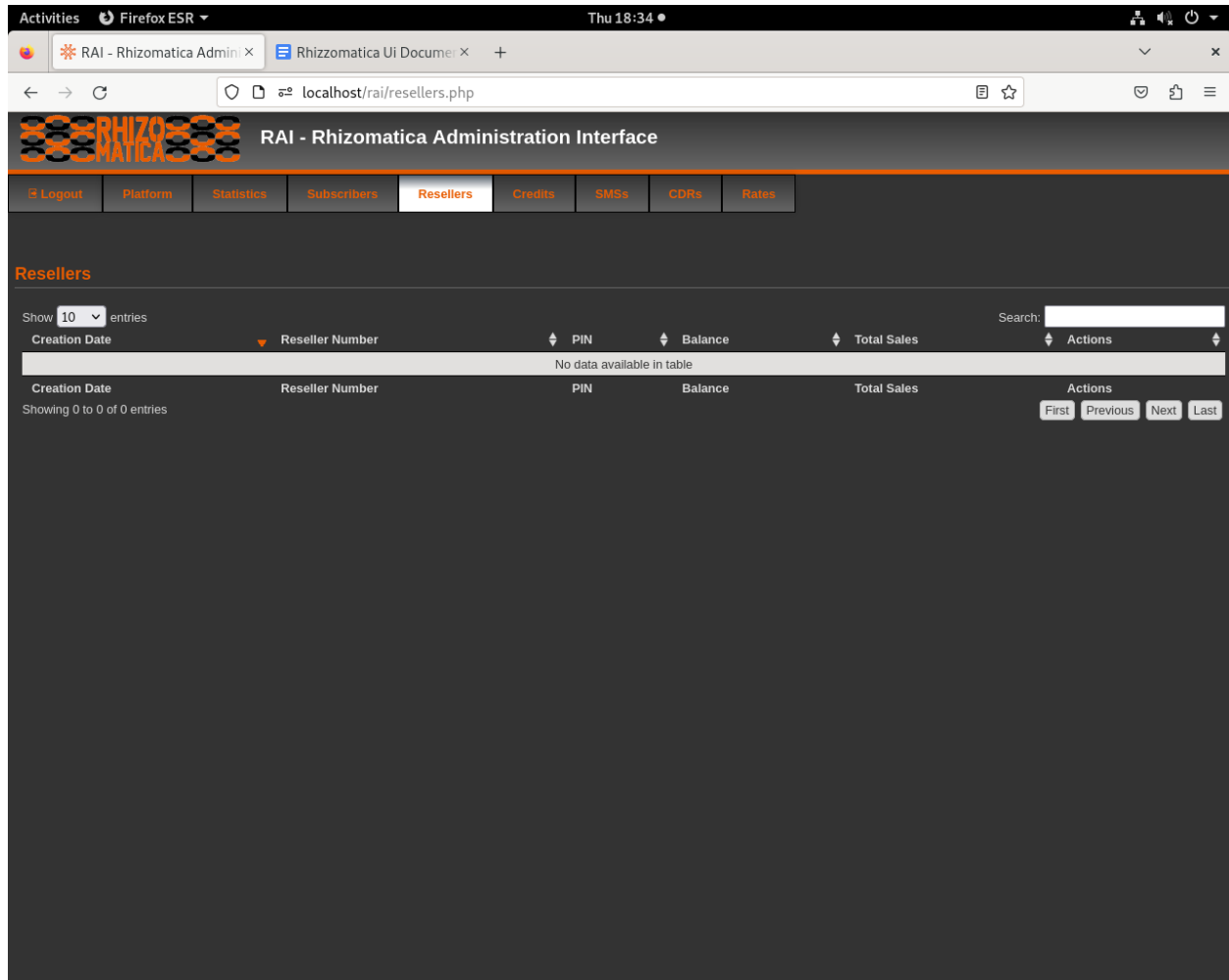


Figure 6.1

Spring 2024

Reseller-provisioning

- This page is contained in resellers_provisioning.php
- Allows the user to add a new reseller by filling out 3 text blocks
- *

The screenshot shows a web browser window with the address bar displaying `localhost/rai/resellers_provisioning.php`. The page title is "RAI - Rhizomatica Administration Interface". The navigation bar includes links for Logout, Platform, Statistics, Subscribers, Resellers (active), Credits, SMSs, CDRs, and Rates. The main content area features a form titled "Provision a new Reseller" with the following fields:

- Reseller number**: A text input field with a label "Reseller number" below it.
- PIN**: A text input field with a label "PIN" below it and "Reseller code" below that.
- Initial Balance**: A text input field with a label "Initial Balance" below it and "Amount to add" below that. The value "0" is entered in the field.

An orange "Add" button is located at the bottom of the form.

Figure 6.2

Spring 2024

Reseller-Add Credit

- This file is contained in resellers_credit.php
- Allows the user to add an add credit by filling out 3 text blocks
- *

The screenshot shows a web browser window with the address bar displaying `localhost/rai/resellers_credit.php`. The page title is "RAI - Rhizomatica Administration Interface". The navigation bar includes links for Logout, Platform, Statistics, Subscribers, Resellers (active), Credits, SMSs, CDRs, and Rates. The main content area features a form titled "Add Credit" with three input fields: "Reseller number" (with a tooltip "Reseller number"), "Reseller number" (with a tooltip "Repeat Reseller Number"), and "Amount" (with a tooltip "Amount to add"). An orange "Add" button is located at the bottom of the form.

Figure 6.3

Spring 2024

Resellers- Invoices

- This page is contained in resellers_credit_history.php
- Reseller Invoices display an empty table to the user as there seems to be no data to display so it cannot be tested. It appears to have similar options to subscribers and probably shares similar issues.

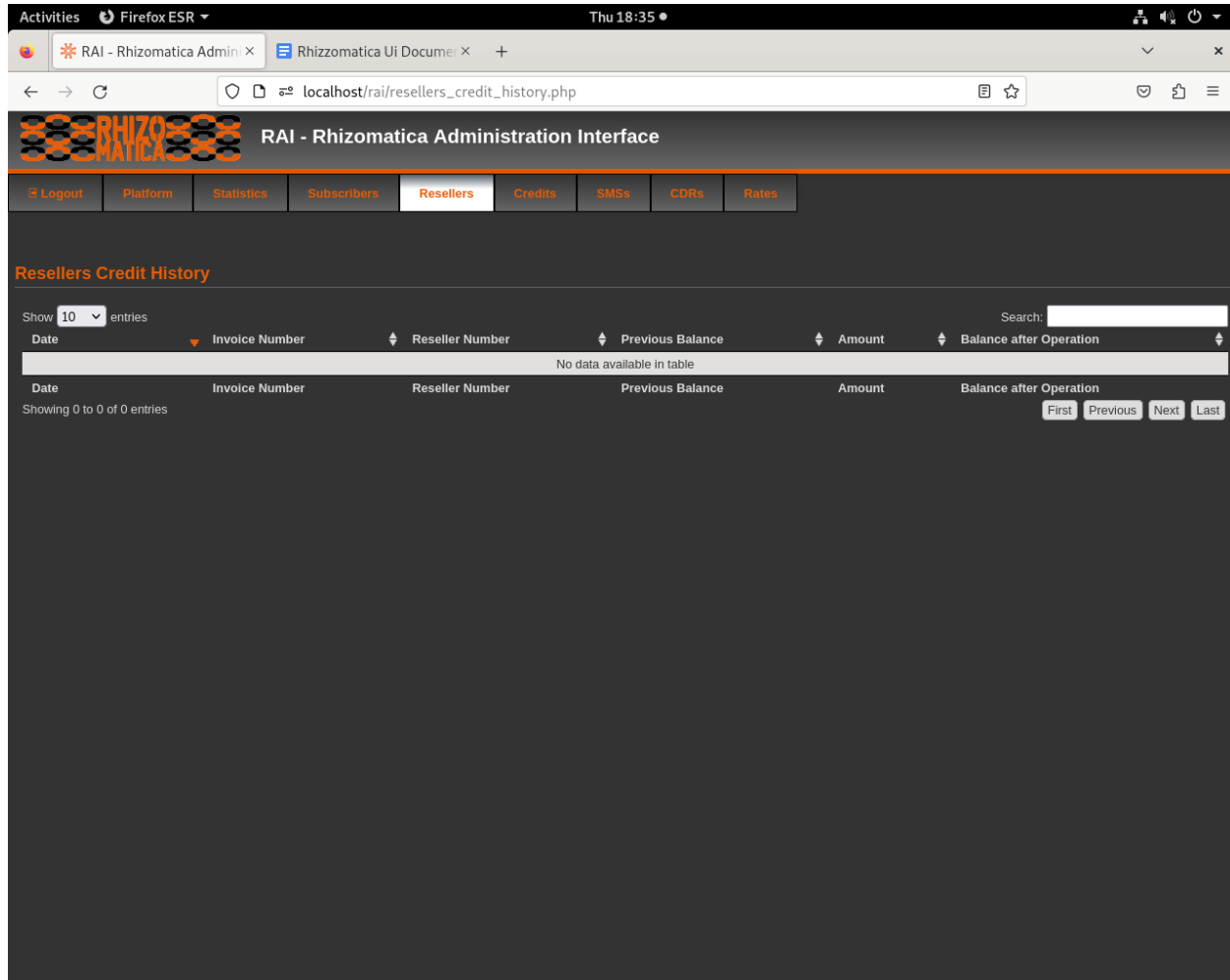


Figure 6.4

Resellers- Transactions

- This page is contained in resellers_transactions.php
- Reseller Transactions displays an empty table to the user as there seems to be no data to display so it cannot be tested. It appears to have similar options to subscribers and probably shares similar issues.

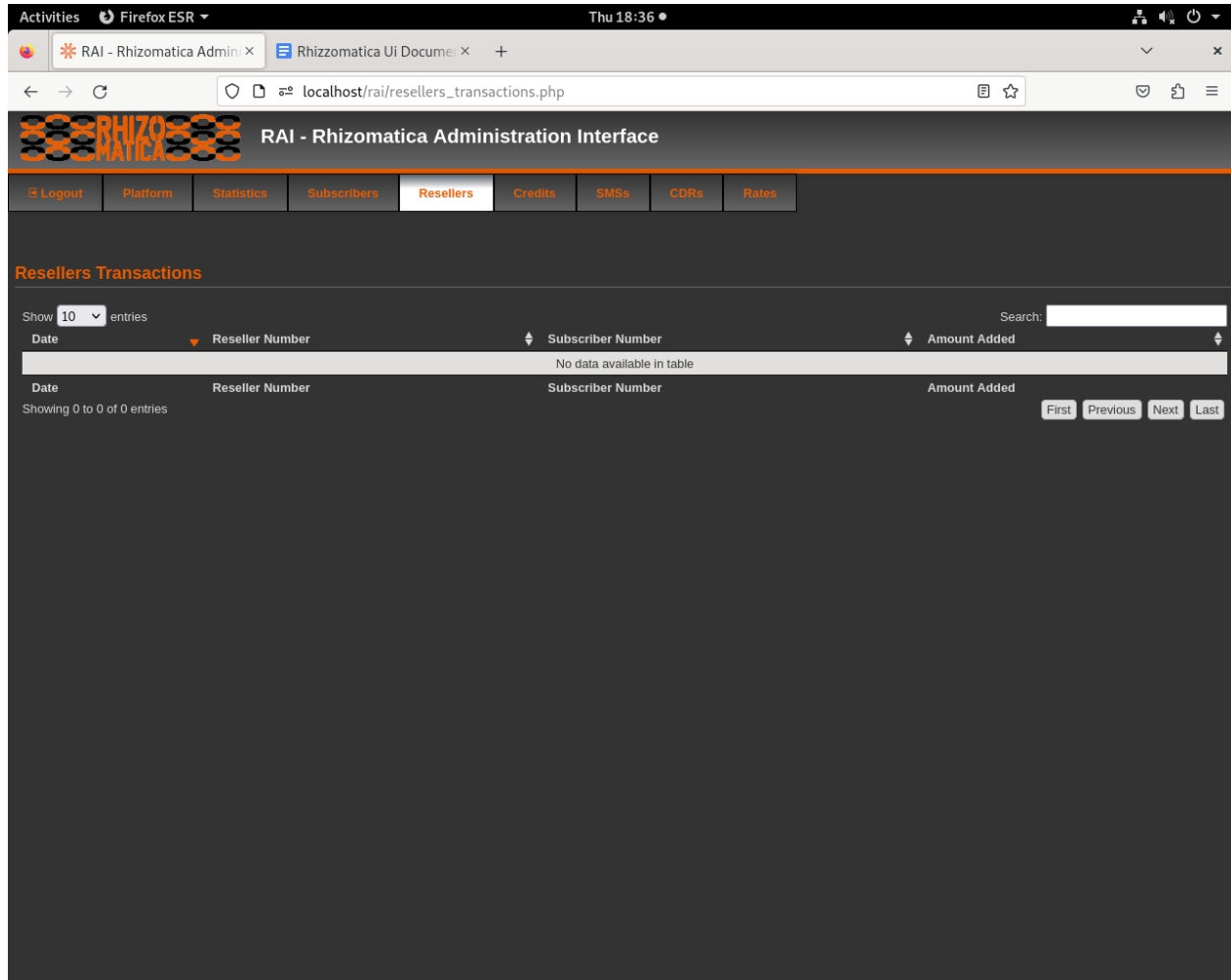


Figure 6.5

Resellers- Configuration

- This page is contained in resellers_configuration.php
- Allows the admin to change notification messages that a reseller might get depending on different conditions, like invalid data, not having enough credits, etc.
- *

The screenshot shows a web browser window with the URL `localhost/rai/resellers_configuration.php`. The page is titled "Configure reseller notification messages" and features a table with six rows for configuring notification messages. Each row has a label for the message type and a corresponding text area for the message content. The messages are:

Message Type	Message Content
Invalid data <small>Invalid data sent to the shortcode</small>	Invalid data
Reseller not enough credit1 <small>Message sent to the subscriber</small>	Reseller does not have enough funds to add credit to your account
Reseller not enough credit2 <small>Message sent to the reseller</small>	Not enough funds to add the credit requested
Credit added successfully to subscriber account <small>Message sent to the subscriber</small>	Amount of [var1] pesos successfully added to your account. New balance: [var2]
Credit added successfully to reseller account <small>Message sent to the Reseller</small>	[var1] pesos successfully transferred to [var3]. Your current balance is: [var4]
General error <small>General error notice message</small>	General error credit could not be added

At the bottom of the form is a "Save" button.

Figure 6.6

Spring 2024

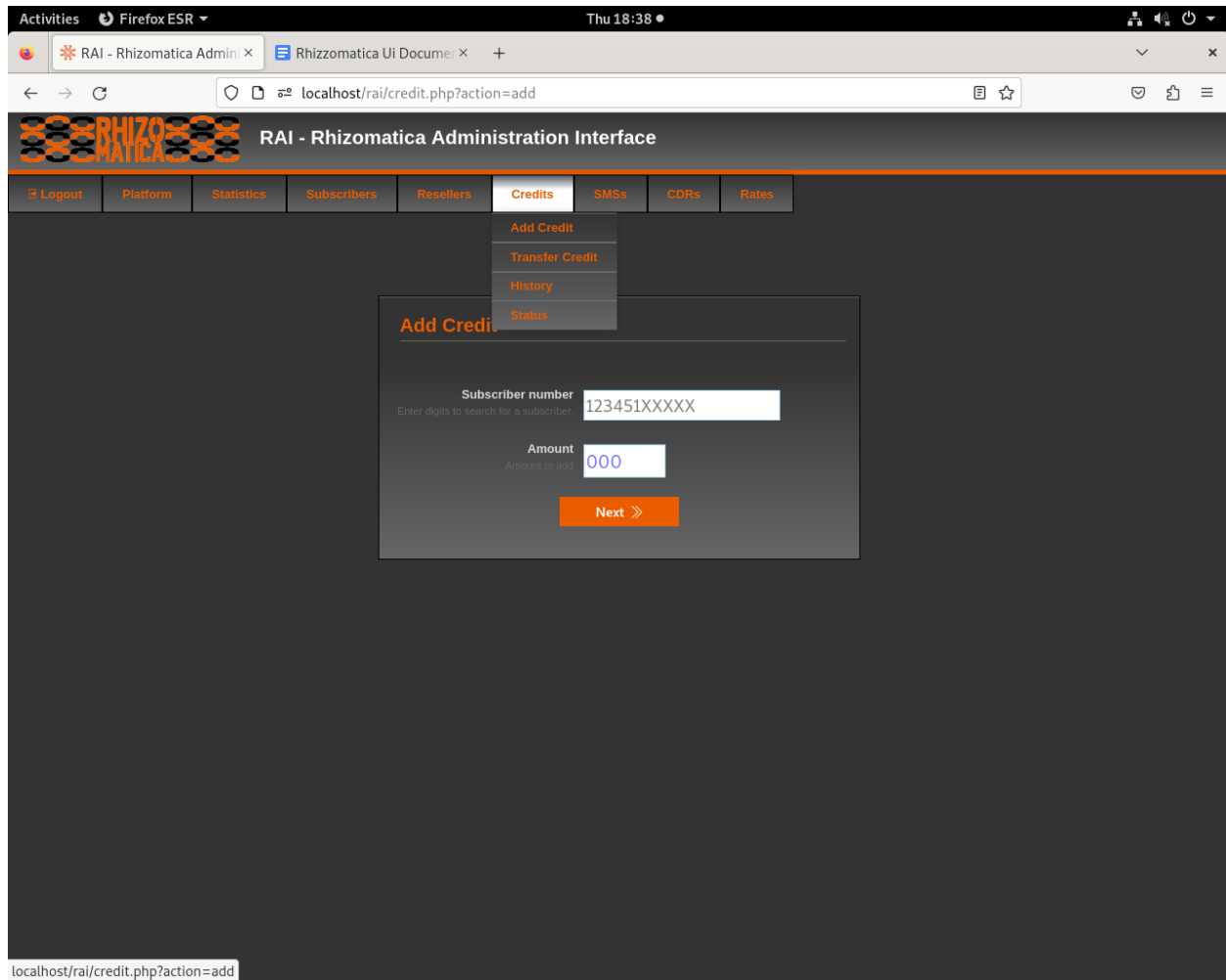


Figure 7.0

Credits - Add Credit

- This page is contained credit.php(Figure 7.1, Figure 7.2)
- Used to either add credits to a subscriber, or move credits from one subscriber to another. Both actions are handled in the same credit.php file, with the parameters being ?action=add or ?action=transfer.
- Allows the user to add a certain number of credits to a subscriber using a subscriber number to add it to a specific customer.
- Do not know what next does as I was told not to touch it

The screenshot shows a web browser window with the address bar displaying `localhost/rai/credit.php?action=add`. The page title is "RAI - Rhizomatica Administration Interface". The interface has a dark grey background with an orange header bar. The header bar contains a logo on the left and a navigation menu with the following items: Logout, Platform, Statistics, Subscribers, Resellers, Credits (highlighted), SMSs, CDRs, and Rates. Below the navigation bar, there is a central form titled "Add Credit". The form has two input fields: "Subscriber number" with the value "123451XXXXX" and a placeholder "Enter digits to search for a subscriber", and "Amount" with the value "000" and a placeholder "Amount in rupees". Below these fields is an orange button labeled "Next »".

Figure 7.1

Credits - Transfer Credit

- Allows the user to transfer a certain number of credits to one subscriber from another using two subscriber numbers to transfer it to the receiving customer.
- Do not know what next does as I was told not to touch it

The screenshot shows a web browser window with the address bar displaying `localhost/rai/credit.php?action=transfer`. The page title is "RAI - Rhizomatica Administration Interface". The interface has a dark grey background with an orange header bar. The header bar contains a logo on the left and a navigation menu with the following items: Logout, Platform, Statistics, Subscribers, Resellers, Credits (highlighted), SMSs, CDRs, and Rates. The main content area is a dark grey box with a white border. Inside this box, the title "Transfer Credit" is displayed in orange. Below the title, there are three input fields: "Source Number" with the value "123451XXXXX", "Destination Number" with the value "123451XXXXX", and "Amount" with the value "000". Each input field has a small hint text below it: "Enter digits to search for a subscriber." for the first two, and "Amount to transfer" for the third. At the bottom of the form is an orange button labeled "Next »".

Figure 7.2

Resellers- History

- Reseller History displays an empty table to the user as there seems to be no data to display so it cannot be tested. It appears to have similar options to subscribers and probably shares similar issues.

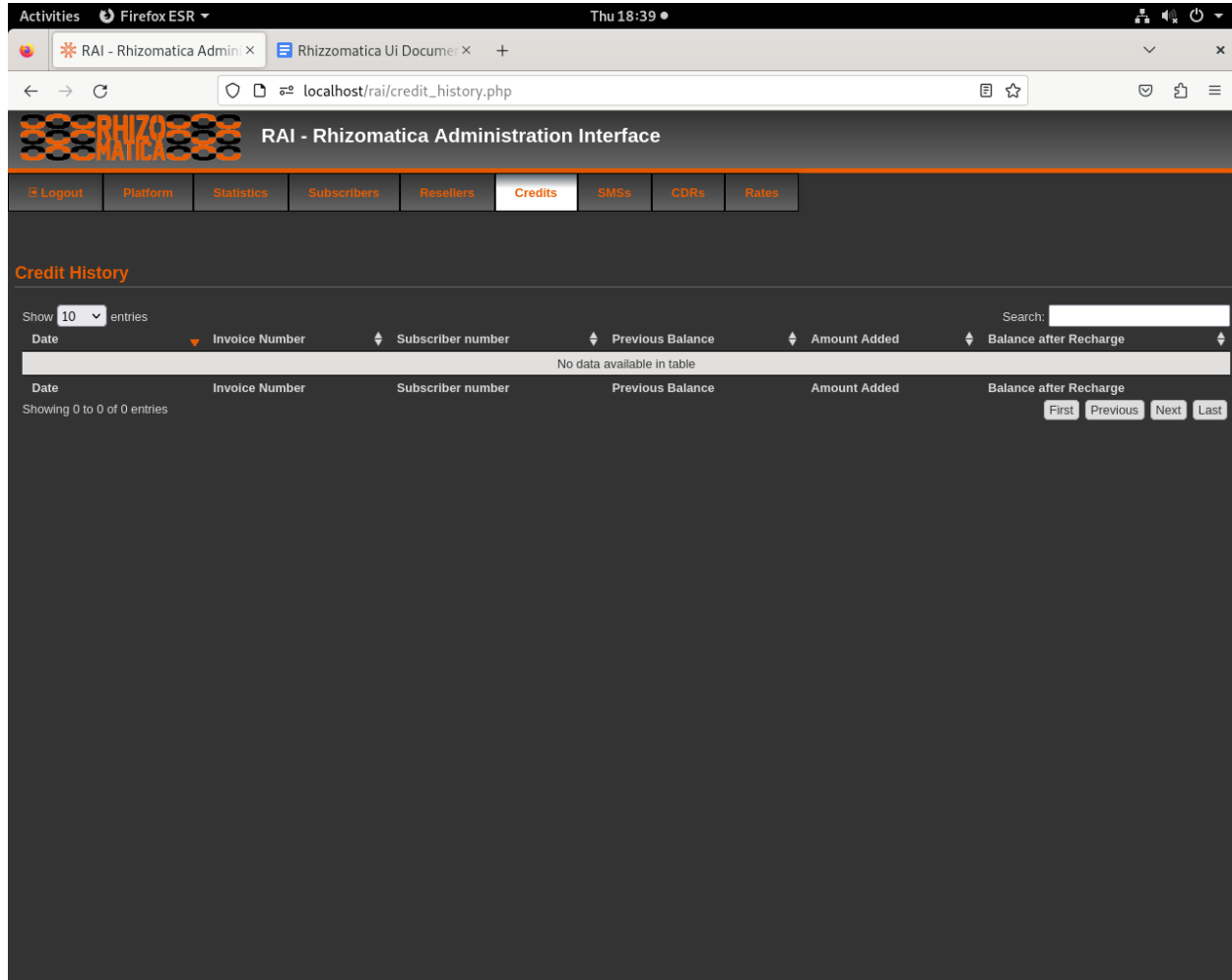


Figure 7.3

Spring 2024

Credit-Status

- This page is completely nonfunctional and none of the drop-down options do anything.

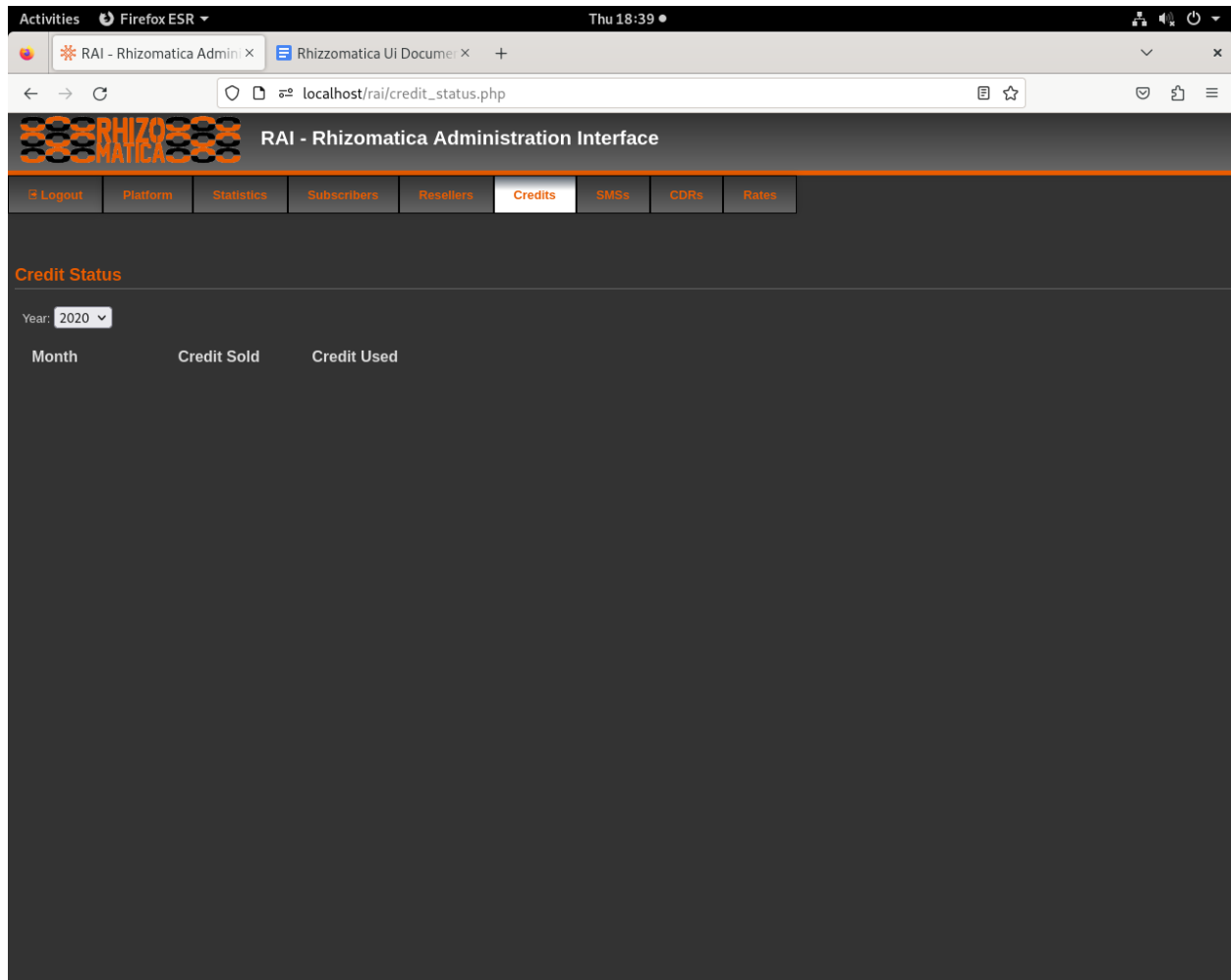


Figure 7.4

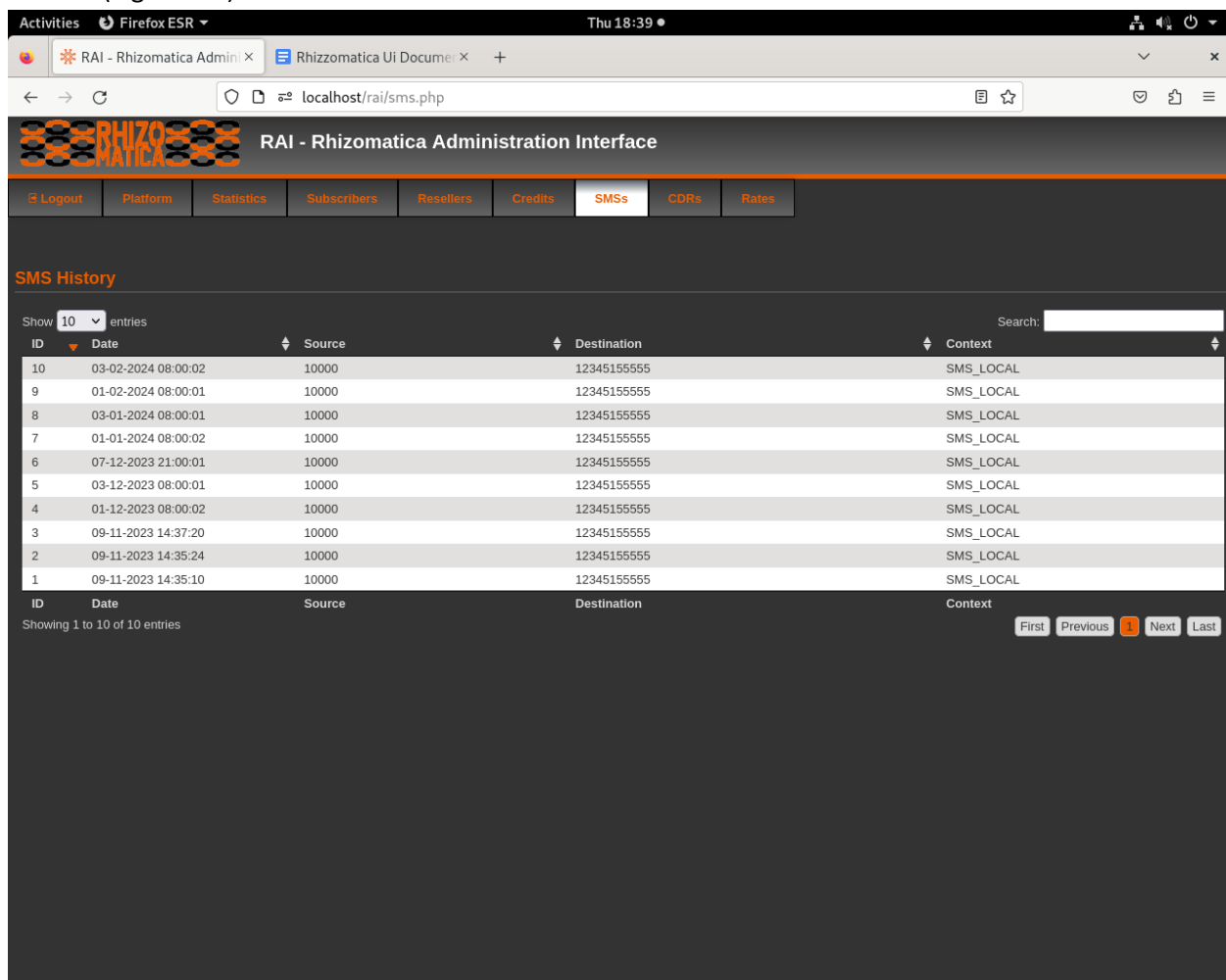
Spring 2024

SMS.php ("rai/SMS.php")

- Not to be confused with the other file with the same name in "modules/SMS.php", which is found in the back-end section of this document.
-

SMSSs

- The drop-down on the top left has the same options as in subscribers and appears to do nothing but this can very much be because of a lack of entries (Figure 8.0)
- The search bar seems to be functioning properly if you try to search for an entry in the fields Source, Destination, and Context, and seems to filter the table correctly (Figure 8.0).
- If you try to look up a date and it exists within the table it will show no document found
- (Figure 8.0.1).
- If you try to look up an ID and it exists within the table it will show no document found or if the number appears in Source or Destination it will presumably try to filter using the numbers found in those fields (Figure 8.0.2).
- The five buttons on the bottom left don't do anything as there are not enough entries in the table (Figure 8.0)



The screenshot shows the RAI - Rhizomatica Administration Interface in a Firefox ESR browser window. The interface has a dark theme with an orange header. The 'SMSs' tab is selected in the top navigation bar. Below the navigation bar, the 'SMS History' section is displayed. It includes a 'Show 10 entries' dropdown, a search bar, and a table with columns: ID, Date, Source, Destination, and Context. The table contains 10 entries, all with Source '10000' and Destination '12345155555'. The Context column shows 'SMS_LOCAL' for all entries. At the bottom, there are pagination buttons: 'First', 'Previous', '1', 'Next', and 'Last'. The status 'Showing 1 to 10 of 10 entries' is also visible.

ID	Date	Source	Destination	Context
10	03-02-2024 08:00:02	10000	12345155555	SMS_LOCAL
9	01-02-2024 08:00:01	10000	12345155555	SMS_LOCAL
8	03-01-2024 08:00:01	10000	12345155555	SMS_LOCAL
7	01-01-2024 08:00:02	10000	12345155555	SMS_LOCAL
6	07-12-2023 21:00:01	10000	12345155555	SMS_LOCAL
5	03-12-2023 08:00:01	10000	12345155555	SMS_LOCAL
4	01-12-2023 08:00:02	10000	12345155555	SMS_LOCAL
3	09-11-2023 14:37:20	10000	12345155555	SMS_LOCAL
2	09-11-2023 14:35:24	10000	12345155555	SMS_LOCAL
1	09-11-2023 14:35:10	10000	12345155555	SMS_LOCAL

Figure 8.0

Spring 2024

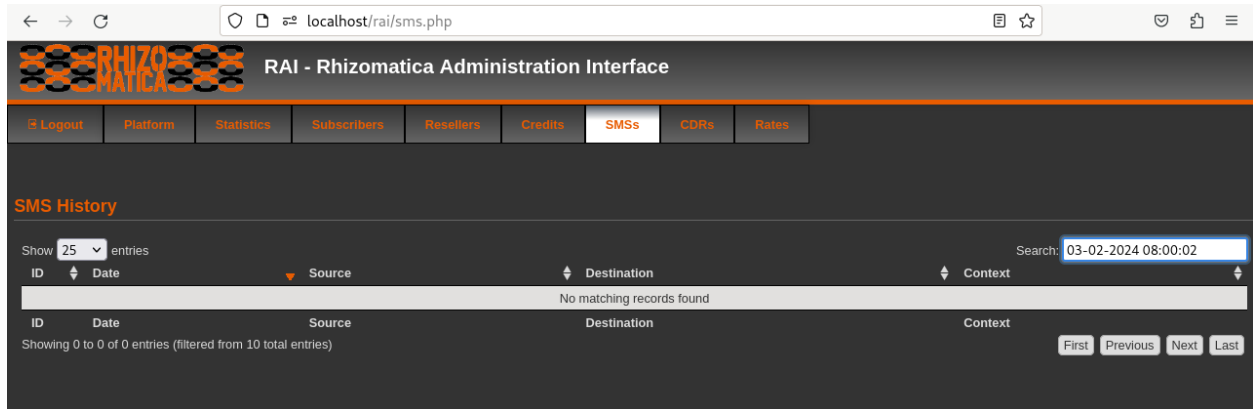


Figure 8.0.1

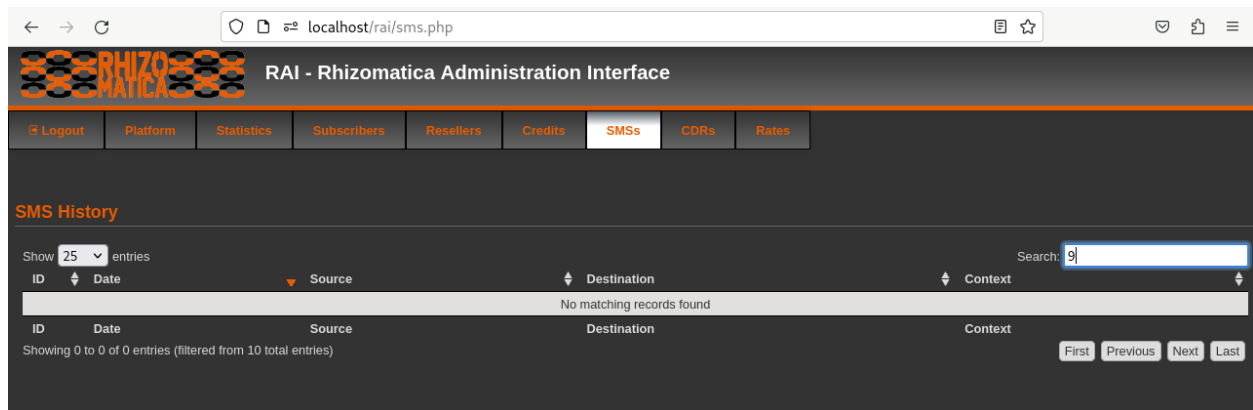


Figure 8.0.2

Spring 2024

SMSs- SendSMS

- Allows the user to send SMS to a specific number and for how long it should try to send the SMS
- For this specific window, we were instructed not to test it, however, its functionality seems to be pretty clear based on the interface shown above. Like with the other files in this section, this is mostly a front-end page, with all of the logic behind it being implemented in the file under "modules/sms.php", which was explained in the back-end section of this document.
- Unsure if this works as I was told not to touch it

The screenshot displays a web browser window with the address bar showing `localhost/rai/send_sms.php`. The page title is "RAI - Rhizomatica Administration Interface". The interface features a top navigation bar with tabs: Logout, Platform, Statistics, Subscribers, Resellers, Credits, SMSs, CDRs, and Rates. The "SMSs" tab is active, and a "Send SMS" button is visible below it. The main content area is a dark gray box titled "Send SMS" containing the following form elements:

- Message:** A text input field with the placeholder "Text of the SMS message".
- Number:** A text input field with the placeholder "Recipient of the SMS message".
- Send SMS:** An orange button.
- Bulk SMS:** A link.
- Validity:** A dropdown menu set to "One Day" with the placeholder "How long to try to deliver SMS".
- Filters:** Four checkboxes for "All Connected", "All Authorized", "All not authorized", and "All subscription not paid".
- Send SMS:** A second orange button at the bottom.

The browser's status bar at the bottom left shows `localhost/rai/send_sms.php`.

Figure 8.1

CDRs

- Probably used to easily identify failures when a call is occurring, since it seems to have a column called "Hangup Cause", which seem to contain some helpful messages, like UNALLOCATED_NUMBER.
- The drop-down on the top left has the same options as in subscribers and appears to do nothing but this can very much be because of a lack of entries (Figure 9.0)
- The check mark button "only calls with cost" makes it so no records appear on the table this makes sense as there is no record with a cost.
- I cannot test any of the functionality with destination and cost as there are no entries that contain those two fields.
- The case does not matter when filtering using the search bar
- The search bar works as expected when looking up values in the following fields Caller Number, Context, and hangup cause looking at any of these values will properly filter the table.
- The search bar does not work for ID, Call Date, Called number, and duration trying to filter with any of the values in these fields will result in no matching records found.
- There are input boxes for each field underneath the table which allows the user to filter using only the values in one field the fields that work are the Caller Number, Called number, Context, and Hangup Cause these all function as expected
- However using the input boxes for ID, Call Date or Duration will do nothing even if the value exists in the table (Figure 9.0.1)
- The five buttons on the bottom left don't do anything as there are not enough entries in the table.

RAI - Rhizomatica Administration Interface

Logout Platform Statistics Subscribers Resellers Credits SMSs CDRs Rates

Calls Details Records

Show 10 entries

Only Calls with Cost ☐ Search:

ID	Call Date	Caller Number	Called Number	Context	Duration	Hangup Cause	Destination	Cost
10	09-11-2023 14:47:57	12345155555	+19152596954	OUTBOUND	00:00:00	NORMAL_CLEARING		
9	09-11-2023 14:43:37	12345155555	0012251004	public	00:00:00	UNALLOCATED_NUMBER		
8	09-11-2023 14:37:19	12345155555	888	EXTEN	00:00:06	NORMAL_CLEARING		
7	09-11-2023 14:35:23	12345155555	888	EXTEN	00:00:05	NORMAL_CLEARING		
6	09-11-2023 14:35:09	12345155555	888	EXTEN	00:00:06	NORMAL_CLEARING		
5	09-11-2023 14:31:10	12345155555	100	EXTEN	00:00:14	NORMAL_CLEARING		
4	09-11-2023 14:29:36	12345155555	100	EXTEN	00:00:02	NORMAL_CLEARING		
3	09-11-2023 14:29:25	12345155555	100	EXTEN	00:00:02	NORMAL_CLEARING		
2	09-11-2023 14:26:47	12345155555	100	public	00:00:00	ORIGINATOR_CANCEL		
1	09-11-2023 14:17:44	12345155555	100	public	00:00:00	CALL_REJECTED		

ID Call Date Caller Number Called Number Context Duration Hangup Cause Destination Cost

Showing 1 to 10 of 10 entries

First Previous 1 Next Last

Figure 9.0

Spring 2024

Activities Firefox ESR Thu 22:01

Rhizzomatica Ui Document RAI - Rhizomatica Admin

localhost/rai/cdr.php

RAI - Rhizomatica Administration Interface

Logout Platform Statistics Subscribers Resellers Credits SMSs CDRs Rates

Calls Details Records

Show 10 entries Only Calls with Cost Search:

ID	Call Date	Caller Number	Called Number	Context	Duration	Hangup Cause	Destination	Cost
6	09-11-2023 14:35:09	12345155555	888	EXTEN	00:00:06	NORMAL_CLEARING		
7	09-11-2023 14:35:23	12345155555	888	EXTEN	00:00:05	NORMAL_CLEARING		
8	09-11-2023 14:37:19	12345155555	888	EXTEN	00:00:06	NORMAL_CLEARING		
10	09-11-2023 14:47:57	12345155555	+19152596954	OUTBOUND	00:00:00	NORMAL_CLEARING		
1	09-11-2023 14:17:44	12345155555	100	public	00:00:00	CALL_REJECTED		
2	09-11-2023 14:26:47	12345155555	100	public	00:00:00	ORIGINATOR_CANCEL		
3	09-11-2023 14:29:25	12345155555	100	EXTEN	00:00:02	NORMAL_CLEARING		
4	09-11-2023 14:29:36	12345155555	100	EXTEN	00:00:02	NORMAL_CLEARING		
5	09-11-2023 14:31:10	12345155555	100	EXTEN	00:00:14	NORMAL_CLEARING		
9	09-11-2023 14:43:37	12345155555	0012251004	public	00:00:00	UNALLOCATED_NUMBER		

9 09-11-2023 14:29:36 Caller Number Called Number Context 00:00:14 Hangup Cause Destination Cost

Showing 1 to 10 of 10 entries First Previous 1 Next Last

Figure 9.0.1

Rates

- This file is very simple, all it does is display the cost for making a call to a certain destination, which at the moment is only either US or Mexico.
- The drop-down on the top left has the same options as in subscribers and appears to do nothing but this can very much be because of lack of entries (Figure 10.0).
- The search function works as expected trying to search for any value will filter the table correctly (Figure 10.0).
- The five buttons on the bottom left don't do anything as there are not enough entries in the table (Figure 10.0).

RAI - Rhizomatica Administration Interface

Logout Platform Statistics Subscribers Resellers Credits SMSs CDRs Rates

Rates

Show 10 entries

Destination	Area Code	Cost (MXN)
Mexico	52	0.28
Us (Estados Unidos)	1	0.18

Showing 1 to 2 of 2 entries

Search:

Cost (MXN)

First Previous 1 Next Last

Figure 10

Back-end

These files are mostly for back-end logic:

- login.php (not a back-end file, but makes explaining some other back-end files easier)
- Session.php
- Configuration.php
 - ConfigurationException
 - Configuration
- sms.php (modules/sms.php)
 - SMSEException
 - SMS
 - send()
 - send_broadcast()

More information about them in their respective sections below.

Session.php

Session.php contains three classes in total:

- AccessManagerException
- AccessManager
- Session

This file also calls the function `session_start()`, which allows the program to use the variable `$_SESSION`, a global array that can store information about the session and can be used in any page where the session was started.

AccessManagerException class

A class that doesn't do anything on its own, as it simply inherits from the already-existing Exception class. Interestingly, this class is not used in session.php nor login.php, so it seems to be useless at the moment.

AccessManager class

Its primary-goal is to handle user authentication, the login and logout logic, credential validation, and initializing the session.

It seems a little bit strange to handle all of those things in a single class, so this entire class could be refactored into different, but more specific classes (due to how coupled the entire system is, it would probably also involve modifying login.php and possibly others, but this seems quite doable). It seems like this class was implemented in such a way that each AccessManager object is treated as an individual user, rather than an actual manager for ALL users.

Attributes:

This class contains five attributes:

- userid
- username
- password
- role
- lang

Out of those five, only **username**, **password**, and **lang** are used. The attribute **userid** could probably be implemented somehow, even if it's not currently used.

Functions:

This class contains five functions:

- login()
- checkPwd()
- initializeSession()
- checkAuth()
- logout()

Login()

```
44  ✓ public function login($username, $password, $lang) {  
45      $this->username = $username;  
46      $this->password = $password;  
47      $this->lang = $lang;  
48  ✓      if ($this->checkPwd($username,$password)) {  
49          $this->initializeSession();  
50  ✓          if (isset($_SESSION['login_goto_uri']) &&  
51              !strstr($_SESSION['login_goto_uri'], 'login.php')) {  
52              header('Location: '.$_SESSION['login_goto_uri']);  
53  ✓          } else {  
54              header('Location: subscribers.php');  
55          }  
56          return true;  
57  ✓      } else {  
58          return false;  
59      }  
60  }
```

What it does

Long story short, it handles the login logic.

I thought that it was worth mentioning this implementation to learn what is good about it and to avoid making the same mistakes in our own version. The login function simply takes three parameters: \$username, \$password, and \$lang, then it stores them in their corresponding class attributes (lines 45, 46, and 47 from the picture above), and finally, if the credentials are properly validated (line 48), it initializes a session and then redirects the user to either the page where they were (lines 50 through 52) or to the subscribers.php page if they come from the login.php page (lines 53 and 54). If successful, the function returns true, otherwise it returns false.

This function performs these function calls:

- `checkPwd()`: Function created in this same file. As the name suggests, it validates whether the username and password are valid. More details on the `checkPwd()` section below.
- `initializeSession()`: Function created in this same file. As the name suggests, it initializes the session. More details on the `initializeSession()` section below.
- `isset()`: PHP function that checks if a variable is set and is not null. More details on the official documentation.
- `strstr()`: PHP function that finds the first occurrence of a string within another string. More details on the official documentation.

For example, if the user was viewing a page but then suddenly it got logged out, then logging back in would redirect it to the page where it was, and if the user came from the login page, then it would get redirected to the `subscribers.php` page, and then the function would return `true`.

What's wrong with it

An improvement that could be applied to `login()` would be to instead of redirecting everything directly within this function, we could either create a `redirect()` function that handles this, or maybe even create an entire different class to handle these actions, which would make it more readable than having to constantly call the `header()` function. For scalability purposes, it's probably better to implement a different class.

Another thing worth mentioning is that the function call to `initializeSession()` on line 49 can probably be removed, and instead, the contents from that function can be directly put here. The reasons for this are:

- `initializeSession()` only contains 3 lines of code and takes no arguments
- As the name of the function implies, this function is only called once
- Removing this useless function would greatly improve readability, since the reader won't have to jump from place to place just to read 3 lines of code.

Overall, this function seems properly implemented on its own, with just the small remarks stated above.

`checkPwd()`

```
63  ✓ public function checkPwd($username,$password) {
64  ✓     require_once(dirname(__FILE__).'../include/database.php');
65      $db_conn = pg_connect(
66      " host=".$DB_HOST.
67      " dbname=".$DB_DATABASE.
68      " user=".$DB_USER.
69      " password=".$DB_PASSWORD);
70      $result = pg_query("SELECT * from users WHERE username='".$pg_escape_string($username)."'");
71  ✓     if (!$result) {
72      |         return false;
73      |     }
74      $row = pg_fetch_row($result);
75      $res = false;
76  ✓     if (password_verify($password, $row[2])) {
77      |         $res = true;
78  ✓     } else {
79      |         $res = false;
80  ✓     }
81      pg_free_result($result);
82      pg_close($db_conn);
83
84      return $res;
85  }
```

What it does

checkPwd() simply validates the username with the given password by performing a query to find the username, and then it uses password_verify() to check if the password corresponds to the user in the database.

What's wrong with it

The checkPwd() function seems to have been coded correctly by implementing the function pg_escape_string(), which mitigates the risk of injection attacks, however, it seems like it could be improved by using prepared statements, a change that would be relatively simple by obtaining the DSN, USER, and PASSWORD from the /include/database.php file to be able to properly use the PDO class. We don't have a variable called DSN, however, that variable is obtained from the database name and hostname, which we have in that file.

See improved function below (it has NOT been tested due to issues in the installation process, but this provides a general idea of how we could improve on the current function by using PDO):

```
// NOT TESTED!!!! JUST A GENERAL OVERVIEW
public function checkPwd($username, $password) {
    require_once(dirname(__FILE__).'/../include/database.php');

    try {
        // Source for line of code below
        // https://www.php.net/manual/en/class.pdo.php
        $dsn = sprintf( 'mysql:dbname=%s;host=%s', $DB_DATABASE, $DB_HOST);

        $dbh = new PDO($dsn, $DB_USER, $DB_PASSWORD);
        $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        // Prepare statement with placeholders
        $stmt = $dbh->prepare("SELECT * FROM users WHERE username = :username");

        // Bind value to placeholder
```

```
$stmt->bindValue(':username', $username, PDO::PARAM_STR);

// Execute the statement
$stmt->execute();

// Fetch the result
$row = $stmt->fetch(PDO::FETCH_ASSOC);

// Check if password matches
if ($row && password_verify($password, $row['password'])) {
    return true;
}
} catch (PDOException $e) {
    // Handle error
    error_log('Connection failed: ' . $e->getMessage());
}
return false;
}
```

Code fragment 1. Source: <https://www.php.net/manual/en/class.pdo.php>

initializeSession()

```
88  public function initializeSession() {
89      $_SESSION['username'] = $this->username;
90      $_SESSION['lang'] = (strlen($this->lang) > 2 ) ? $this->lang.'.utf8' : $this->lang."_".
          strtoupper($this->lang);
91      $_SESSION['is_logged'] = 1;
92  }
```

What it does

Sets the values of the global attribute **\$_SESSION** to the corresponding **username** and **lang** values, and uses the Boolean variable **is_logged** to indicate that a session has started (1 means true, 0 means false).

What's wrong with it

This function is only used once, takes no parameters (everything used inside of it is global), and it's only three lines of code. Seems kind of pointless to create a function for this purpose, and,

arguably, it makes reading the code more difficult by adding the need to jump from place to place to know what is being performed.

I believe that a much better approach would have been to make it take the corresponding parameters, and make the changes based on those parameters. As mentioned at the beginning of the AccessManager class section, this implementation seems to be treating the class similar to how an individual user would be treated, which makes no sense, since the class name suggests that this is supposed to be the manager for all users.

checkAuth()

```
94  ✓ public function checkAuth() {  
95  ✓     if (!isset($_SESSION['username']) && !isset($_SESSION['is_logged'])) {  
96         $_SESSION['login_goto_uri']=$_SERVER["REQUEST_URI"];  
97         header('Location: login.php');  
98     }  
99 }
```

What it does

Checks that there's a valid logged-in user in the session, and redirects to the **login** page if it fails.

If everything goes well, the function does nothing and the program continues.

What's wrong with it

On the way it was implemented, there's nothing wrong with it, but as mentioned earlier, it's preferable to make functions work with given parameters rather than with global variables for both readability and modularity. Regardless, even if we create the AccessManager class from scratch, we still need to accomplish what this function does, so this function would still be very useful and would require minimum changes.

logout()

```
101  ✓    public function logout() {  
102        unset($_SESSION['username']);  
103        unset($_SESSION['lang']);  
104        unset($_SESSION['is_logged']);  
105        header('Location: login.php');  
106    }
```

What it does

Similar to the `initializeSession()` function, this function is very simple (in fact, this function is even simpler, due to the fact that it just resets the values of the `$_SESSION` variable). This is the only function that seems to not require any changes, so it's worth keeping an eye on this one for our own implementation.

What's wrong with it

On it's own, nothing, the only change needed would be in which class to put this function, which has not been planned.

Session class

The session class only contains the attribute `$access`, which is used to instantiate an `AccessManager` object when a session object is instantiated, which, as explained above, handles multiple things at once, so it seems a little strange that this class was added in the first place. Regardless, this class has the `store()` function which applies changes to the previously mentioned `$_SESSION` variable, so that the information can be properly accessed from other pages within the same session.

Summary of Session.php

The main problem with this file is that what it accomplishes is not that complicated, but it was extremely difficult to read and understand due to all the things that required me to hop from file to file. The improvements of this file seem to be relatively straight forward and quite doable, but it does require some planning beforehand so that we don't make the same mistake of making it almost impossible to read. Some of the functions in this file are pretty decent and usable by themselves, the only problem was how they were organized.

Another thing worth mentioning is that it is extremely likely that this file is only used on the login.php file, which is very good because it means that even if we were required to build this part of the system from scratch, it wouldn't be completely impossible.

Main points for Session.php:

1. Everything is very tightly coupled
2. checkPwd() was good, but could be better by using prepared statements to improve security
3. header() is called multiple times in the file, but it would be much better to either create an entire new class or a new function in an existing class to handle the redirects
4. initializeSession() seems to be useless, and it negatively affects readability
5. The parameter **userid** is not implemented in the login, but it probably should. We could modify it so that a proper **userid** is implemented (this is not necessary at the moment, but it would be good to have it already implemented in case it's needed in the future)
6. There are barely any comments in the code

7. There aren't any Exceptions thrown in this entire file, so we need to ensure that all possible errors are properly handled
8. The functions seem to be all over the place, and some functions don't seem to be consistent with the classes where they were put

Configuration.php

Contains the following classes:

Configuration

The configuration class contains one private attribute that contains a path, and four functions that perform HTTP GET requests that attempt to retrieve informative files in JSON format from said path.

The path variable is set to **"http://localhost:8085/configuration"**

All four functions accomplish the same thing, with the only difference being what they are attempting to retrieve. These functions are:

- `getSite()`

```
12     public function getSite() {  
13         try {  
14             $response = \Httpful\Request::get($this->path."/site")->expectsJson()->send();  
15             $data = $response->body;  
16         } catch (\Httpful\Exception\ConnectionErrorException $e) {  
17             throw new SubscriberException($e->getMessage());  
18         }  
19         if (!is_array((array) $data)) {  
20             if ($data->status == 'failed') {  
21                 throw new SubscriberException($data->error);  
22             }  
23         } else {  
24             return $data;  
25         }  
26     }
```


- getConfig()

```
28     public function getConfig() {
29         try {
30             $response = \Httpful\Request::get($this->path."/config")->expectsJson()->send();
31             $data = $response->body;
32         } catch (\Httpful\Exception\ConnectionErrorException $e) {
33             throw new SubscriberException($e->getMessage());
34         }
35
36         if (!is_array((array) $data)) {
37             if ($data->status == 'failed') {
38                 throw new SubscriberException($data->error);
39             }
40         } else {
41             return $data;
42         }
43     }
44 }
```

- getPackages()

```
45     public function getPackages() {
46         try {
47             $response = \Httpful\Request::get($this->path."/packages")->expectsJson()->send();
48             $data = $response->body;
49         } catch (\Httpful\Exception\ConnectionErrorException $e) {
50             throw new SubscriberException($e->getMessage());
51         }
52
53         if (!is_array((array) $data)) {
54             if ($data->status == 'failed') {
55                 throw new SubscriberException($data->error);
56             }
57         } else {
58             return $data;
59         }
60     }
61 }
```

- getLocations()

```
62  public function getLocations() {
63      try {
64          $response = \Httpful\Request::get($this->path."/locations")->expectsJson()->send();
65          $data = $response->body;
66      } catch (\Httpful\Exception\ConnectionErrorException $e) {
67          throw new SubscriberException($e->getMessage());
68      }
69
70      if (!is_array((array) $data)) {
71          if ($data->status == 'failed') {
72              throw new SubscriberException($data->error);
73          }
74      } else {
75          return $data;
76      }
77  }
```

All four functions above are exactly the same, except for the GET request, which changes depending on what we want to retrieve, so a very obvious improvement is to create a single function that accomplishes the same thing as the four functions above, where we simply pass what we want to retrieve as a parameter.

As their name suggests, all functions simply expect to retrieve information about the site, config, packages, and locations respectively. Failures may occur either when there are connection errors, or when the retrieved information is not in an array format and the status gets set to "failed". In both of these cases, a `SubscriberException` is thrown.

ConfigurationException

Empty class that inherits from the Exception class. As the name suggests, it's used to be thrown when Exceptions occur in the Configuration class.

SMS.php ("modules/SMS.php")

Not to be confused with the other file with the same name in "*rai/SMS.php*", which is found in the front-end section of this document.

Similar to Session.php and Configuration.php, this file is used by other front-end files. It contains two small classes:

SMSException

Empty class that inherits from the Exception class. As the name suggests, it's used to be thrown when Exceptions occur in the SMS class.

SMS

Class that contains a private attribute (which is set to "**http://localhost:8085/sms**", and two functions responsible for sending messages. One function sends it to a single number, and the second one sends a broadcast. Both of the functions below seem to be short, and they seem to properly handle any validation needed or any Exceptions that might occur.

send()

Performs an HTTP POST request with the data being an array with the **source, destination, and message to send** (in that order). It then catches all the appropriate Exceptions and validates the status of the response.

```

12  ✓ public function send($source,$destination,$text) {
13      $data = array("source" => $source, "destination" => $destination, "text" => $text);
14  ✓      try {
15          $response = \Httpful\Request::post($this->path."/send")->body($data)->sendsJson()->send();
16  ✓      } catch (\Httpful\Exception\ConnectionErrorException $e) {
17          throw new SMSEException($e->getMessage());
18      }
19  ✓      if ($response->code != 201) {
20          throw new SMSEException($response->body);
21      }
22      $data = $response->body;
23  ✓      if ($data->status == 'failed') {
24          throw new SMSEException(htmlentities($data->error));
25      }
26  }

```

send_broadcast()

Similarly to `send()`, it performs an HTTP POST request, but with the array containing **message to send, btype, location, and validity** (in that order). It also catches all the appropriate Exceptions, and it also validates the status of the response.

```

28  public function send_broadcast($text, $btype, $location, $validity) {
29      $data = array("text" => $text, "btype" => $btype,
30                  "location" => $location, "validity" => $validity);
31      try {
32          $response = \Httpful\Request::post($this->path."/send_broadcast")->body($data)->sendsJson()->send();
33      } catch (\Httpful\Exception\ConnectionErrorException $e) {
34          throw new SMSEException($e->getMessage());
35      }
36
37      $data = $response->body;
38      if ($data->status == 'failed') {
39          throw new SMSEException($data->error);
40      }
41      if ($data->status != 'success') {
42          throw new SMSEException($data);
43      }
44  }

```

Sources

Code fragment 1: <https://www.php.net/manual/en/class.pdo.php>

Kiwi IRC: <https://github.com/kiwiirc/kiwiirc/wiki/Example-installation-guide>