CDR PROJECT PART 2
date: 26th Jan 2013.

SUMMARY OF PROJECT TASKS

UPDATES TO CURRENT CDR:

1. Add 'info' button to the current CDR web pages with a ? symbol.
2. Combine 'membership.sql' .Change 3 tables into 1 table & update code to match
3. Confirm 'ignore list' is working as '*97' does not seem to be ignored always.
4. Add code to display date on web pages
5. variable to control how often code reads the remote MySQL database

NEW FEATURES:

6. Create front web page
7. Create a 'Settings' web page for update of 'membership.sql' database fields.
8. Create 'Call Reports' web page for scheduled emailing of reports.

TASKS IN DETAIL

1: Add 'info' button to the current CDR web pages with ? symbol

- Place 'info' pop-up button in top of all pages using ? Symbol
  (see '***Page updates/Add_Help_Button.png***')
- Do this for all pages using the following text format in the code so I can find it and update the info it later: ***'This is the help pop-up for the xxx web page'***

2: Combine 'membership.sql'. Change 3 tables into 1 table & update code to match.

- Change 'membership.sql' database. **acl_user, acl_group, acl_membership** tables into 1 table containing 3 fields: extension, username, team
- this will require updating the current CDR code to match the change.

3: Confirm 'ignore list' is working as '*97' does not seem to be ignored always.

- In tests this didnt always work when *97 was in the dst field of the asteriskcdrdb database on the remote Mysql , confirm this feature is working.

4: Add code to display date on web pages

- Add code to show the current date on the web pages. This should display in top left of page and be toggle enabled for all pages or no pages on the 'Live Call Display' pages.
e.g. *$DateOn = 1* will show current date on all web pages. *$DateOn = 0* will hide this on all web pages.
- For example see (see '***Page updates/Add_Help_Button.png***')

5: variable to control how often code reads the remote MySQL database

This is needed to control how often the CDR code accesses the remote server. If it overloads the server processes this can cause problems with phone calls. Ideally it will only access the MySQL remote server database every 1-5 seconds.

6: Create front web page

- see example of front page – ***'Page_Updates/Front_Page.png'***

- 'Live Call Display' should link to the current CDR web site page.
- 'Call Reports' should link to the 'Call Reports' page detailed in task # 7.
- 'Settings' should link to the 'Settings' page detailed in task # 6.

7: Create a 'Settings' web page for update of 'membership.sql' database fields.

- This page is for updating data in the local 'membership' database. Fields are: extension, username, & team (see task # 2) .
- See **'Page Updates/Settings_Page.png'** for basic example web page.
- All fields should be loaded to the page from the membership database on Refresh and visible so they can be updated directly in the table window on the web page.
- There should be a scroll-bar feature for larger numbers of records.
- How a user can delete records is up to the coder. Either a delete button next to each record, or clear fields using delete on the keyboard from inside each field.
-

8: Create 'Call Reports' web page for scheduled emailing of reports.

See the web page example: **'page updates/Call_Reports.png'**

*(Also provided are two example html coded reports that have been created as email attahments by CDR software called TeleTellus Pro. Use these as reference only. They are in **/Example_HTML_Reports/**)*

Scheduled reports options should be stored as files in /reports folder. Each file will contain the details of the fields saved from the 'Call Reports' web page. The CDR software should run a check at midnight every day and process any files it finds in the /reports folder. Files that have a matching date should be run. Files that do not contain a matching date should be ignored. When files are run they will send an html attached report (see examples) to the email address supplied in the file, showing all the info based on the fields provided in the file. These files are created using information entered via the 'Call Reports' web page. List of fields are here:

- Load report – This can be a button and should go to /reports folder with a browser window so the user can load a saved report file into the web page display fields.

- **Email Address** – This should be a single email address that the report will be sent to.

- Extension list – This is a list of Extensions to run the report for. Use commas to seperate extensions e.g. 500,501,506 . If left blank = all extensions.

- Date Filter – This should be a calendar drop-down list that populates the field and can also be entered manually using DD/MM/YY format. When this field is filled, the 'Schedule' field will be forced to empty.
- **Schedule** – This is a drop down list providing 4 options : Daily, Weekly or Monthly (or empty) When this is selected the 'Date Filter' will be forced to empty.

- Report Type – This is a selection option. Only one report type can be selected.

> **- Call Counts** – This should create a html page report similar to the 'Live Call Display' pages (using the same variables & calculations) . It should show the following columns using the supplied dates or schedule, extension list filter, and sent as an html attachment to the provided email address:

Column 1: **Extension Number** (database = membership; field: see task #2)
Column 2: **Username** (database = membership; field = see task # 2)
Column 3: **Total number of calls**
Column 4: **Unique dialled numbers**
Column 5: **Total call count < 1 min**
Column 6: **Total call count 1 to 5 mins**
Column 7: **Total count > 5 mins**
Column 8: **Total Call time (hh:mm:ss)**

- There should also be a final row that shows **TOTALS of each column 3,4,5,6,7 & 8**.
- The html should be in descending order based on Extension number.
- There should be only one row per extension number.

- Detailed Report – This should create a html page report showing the following columns using the supplied dates or schedule, extension list filter, and sent as html attachment to the provided email address:

Column 1: **Date & Time** (database = asteriskcdrdb, field = calldate)
Column 2: **Username** (database = asteriskcdrdb, field = clid)
Column 3: **Extension number** (database = asteriskcdrdb, field = src)
Column 4: **Destination Number** (database = asteriskcdrdb, field = dst)
Column 5: **Call duration** (database = asteriskcdrdb, field = billsec)

- There should also be a final row that shows **TOTAL of the column 5.**
- The html should be in descending order based on Date & Time.
- There may be multiple rows of the same extension number in this report.

All reports should have the email subject line: **'CDR Plus scheduled report'**
Each report should be have a title showing the **date range of report**

- Save Report – this can be a button and will open to the /reports folder and allow the user to save the fields entered in the web page to the report. The can use any naming convention but the code should enforce the file type. They should not be able to save outside of the /reports folder.

- Display Report – this can be a button and should open a seperate browser tab showing the html report that would be created by the current fields on the web page. If a field is not filled in it should display a pop-up error saying '**Please make sure all fields on the 'Call Report' page are correctly entered**'

- Send Now – this can be a button and should run the report based on the completed fields in 'Call Report' web page and email it to the address listed in Email Address field. On error it should display: '**Please make sure all fields on the 'Call Report' page are correctly entered**'