Technical User Manual

FOR THE UPGRADRE OF THE GBV/VAC HELPLINE SYSTEM

Submitted By

Table of Contents

1. Introduction

The solution is made up of two main components and a number of dependencies as described here. The two main parts are: Case Management Systems (CMS) and Call Module powered by Asterisk.

Asterisk runs best on Linux based operating systems and Centos 8 is recommended for this solution. All descriptions related to this version shall be based:

- Centos 8
- Nginx
- MySql

2. Case Management System

This is a web solution developed on PHP, html and Javascript. It runs on a web (apache/nginx/httpd) server and uses MySql database.

To setup the solution, install the following and their dependencies. It is advisable to run the installations as a non-root user and have firewall running.

2.1 Install a web server

In our case, we will use Nginx web server

In order to install Nginx, we'll use the `dnf` package manager, which is the new default package manager on CentOS 8.

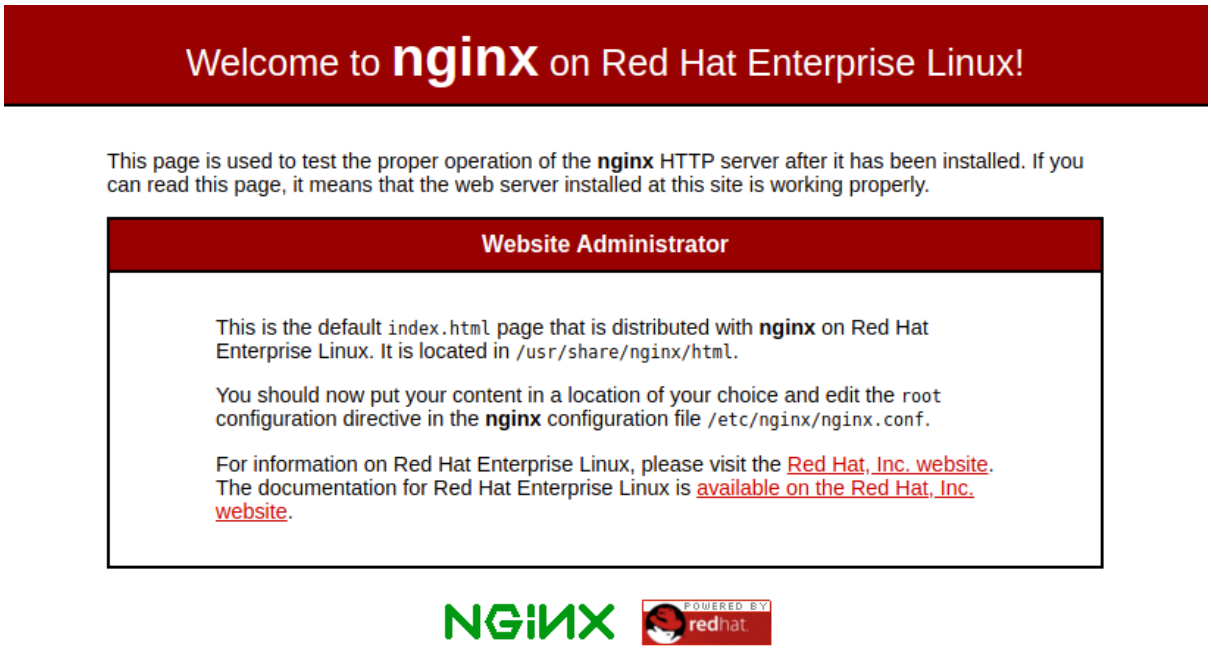
Install the `nginx` package with:

```
sudo dnf install nginx
```

Enable and start the server:

```
sudo systemctl enable nginx
sudo systemctl start nginx
```

Check on the browser with your domain IP or using 127.0.0.1 if on current computer. The service runs on port 80 by default. You should be able to get a Nginx welcome page as below if the server is correctly and successfully installed.



Configuring Nginx

Nginx can host several web application in different drectories and different domains. Since we have one application for our case, we will install our application on Nginx root folder. The default Nginx root directory is

```
/usr/share/nginx/html
```

Most users prefer changing the root directory to:

```
/var/www/html
```

Nginx configuration files are found in `/etc/nginx/` directory and the default configuration file is `/etc/nginx/nginx.conf`. The root directory can be changed on the configuration file.

Another important Nginx directory is `/etc/nginx/conf.d/`, this directory contains server block configuration files, where you can define the websites that are hosted within Nginx. A typical approach is to have each website in a separate file that is named after the website's domain name, such as `helpline.conf`.

For our case, we will have the application server block within the `etc/nginx/nginx.conf` configuration file as below: Update the configuration file with the below lines, some may already exist.

```
server {
        listen 80;
        listen [::]:80;
```

```
        root /var/www/html;
        index index.html index.htm index.nginx-debian.html;

        server_name helpline.sematanzania.org www.helpline.sematanzania.org;

        location / {
                try_files $uri $uri/ =404;
        }
}
```

To make sure that there are no syntax errors in any of your Nginx files, run:

```
sudo nginx -t
```

Once your configuration test passes, restart Nginx to enable your changes:

```
sudo systemctl restart nginx
```

With this and CMS installation, the application will be available on the local IP and the domain on condition that the domain and the local server IP have been mapped to a public IP.

Also, for php to work with nginx, install php-fpm with the following command:

```
sudo systemctl enable php-fpm

sudo systemctl start php-fpm
```

Configure php-fpm listen to php-fpm.sock in /etc/php-fpm.d/www.conf. The default path for the sock file is /var/run/php-fpm/php-fpm.sock

2.2 Install MySQL

Run the following command to install the mysql-server package and a number of its dependencies:

```
sudo dnf install mysql-server
```

With that, MySQL is installed on the server but it isn't yet operational. The package you just installed configures MySQL to run as a systemd service named mysqld.service. In order to use MySQL, there is need to start.

```
sudo systemctl start mysqld.service
```

Then set MySQL to start whenever the server boots up

```
sudo systemctl enable mysqld
```

To secure MySQL, run the following commands and follow the prompts.

```
sudo mysql_secure_installation
```

This will enable one to set a password for the root user, however, it is recommended not to run the system with MySQL root user

2.3 Install PHP

The installed web server can execute files which do not need compilations such as html files, php compiler will need to be installed for php files to be executed on the server.

To install the php and php-mysqlnd packages using the dnf package manager, run:

```
sudo dnf install php php-mysqlnd
```

Once the installation is complete, restart the Nginx server in order to enable the PHP module.

2.4 Install the CMS

This is done by simply pulling the code from the code repository, either the solution github repository or a local directory.

Copy the downloaded files to the nginx server root folder /var/www/html/helpline/

Create a configuration file /var/www/tz_config.php

3. Install and configure Asterisk

Install Asterisk

The current system runs and operates optimally with Asterisk 16 and is recommended to install from source. The following steps describe the process:

Download Asterisk 16 source files from to your server using the following command:

```
wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20-current.tar.gz
```

Other Asterisk dependencies we will install are:

i. Lipri - The libpri library allows Asterisk to communicate with ISDN connections. We will need because of using DAHDI with ISDN interface hardware such as E1. Download Link
ii. Dahdi - The DAHDI library allows Asterisk to communicate with analog and digital telephones and telephone lines, including connections to the Public Switched Telephone Network, or PSTN. Download Link

After downloads, untar the source files using the following commands:

```
tar -zxvf libpri-current.tar.gz

tar -zxvf dahdi-linux-complete-current.tar.gz

tar -zxvf asterisk-20-current.tar.gz
```

Build and install dahdi using the following commands:

```
cd dahdi-linux-complete-current
```

```
make

make install

make config
```

Build and install LibPRI using the following commands:

```
cd libpri-current

make

make install
```

Now compile and install asterisk with the following commands:

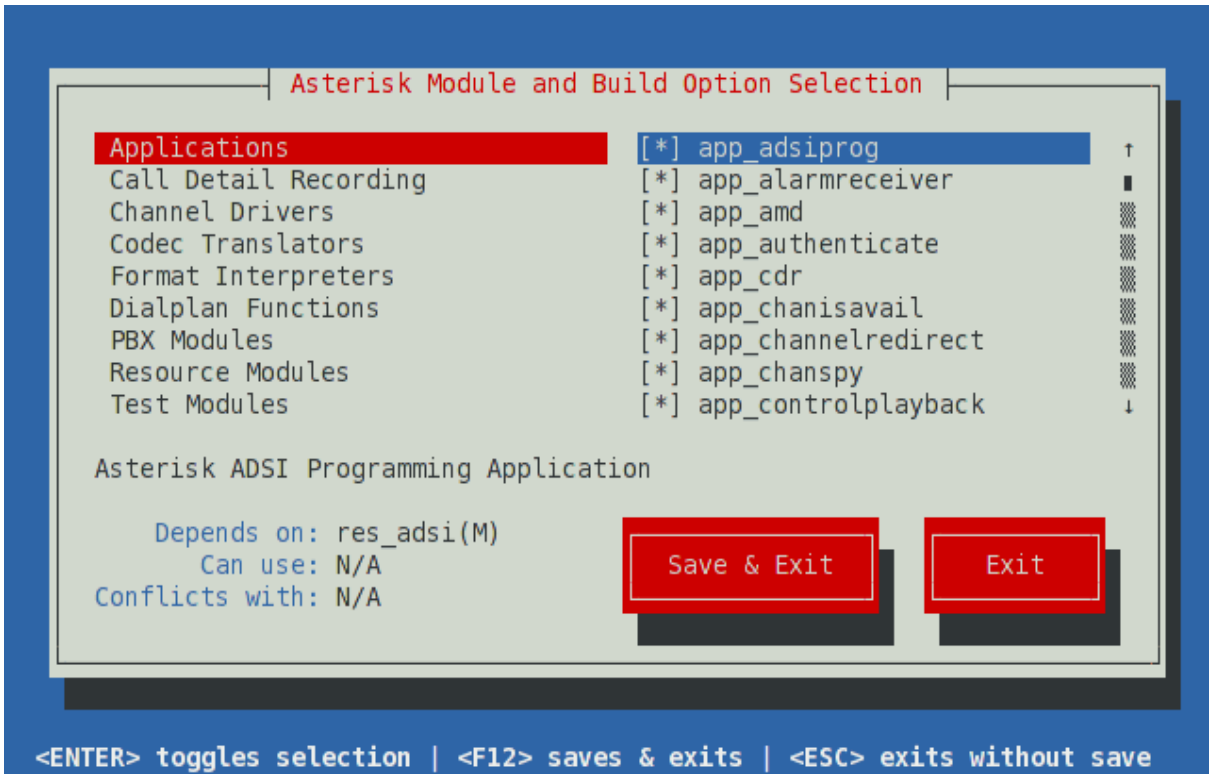
```
cd asterisk-16-current

./configure
```

Resolve any dependency issue that may cause the ./configure to fail then use the following command to select the desired asterisk menu items.

```
make menuselect
```

This command will have a screen like this:



To compile Asterisk, simply type make at the Linux command line.

```
make
```

Then install Asterisk using the following command:

```
make install
```

At this point, Asterisk installation is complete and can start or check the service status using the commands below:

```
/etc/init.d/asterisk start

/etc/init.d/asterisk status
```

Install SSL certificates

Asterisk provides a utility script, ast_tls_cert in the contrib/scripts source directory. We will use it to make a self-signed certificate authority and a server certificate for Asterisk, signed by our new authority.

Still on the Asterisk installation source directory, run the following commands:

```
sudo mkdir /etc/asterisk/keys

sudo contrib/scripts/ast_tls_cert -C helpline.sematanzania.org  -O "Csema Helpline" -b 2048 -d /etc/asterisk/keys
```

Follow the prompts keenly not to skip any necessary information. A successfully certificate generation will create the following files on the certificate directory mkdir /etc/asterisk/keys.

```
asterisk.crt
asterisk.csr
asterisk.key
asterisk.pem
ca.cfg
ca.crt
ca.key
tmp.cfg
```

Configuring Asterisk

To meet the system requirements for optimal operations, follow the following configuration steps with the respective parameters

To communicate with WebSocket clients, Asterisk uses its built-in HTTP server. Configure `/etc/asterisk/http.conf` as follows:

```
[general]
enabled=yes
bindaddr=0.0.0.0
bindport=8088
tlsenable=yes
tlsbindaddr=0.0.0.0:8089
tlscertfile=/etc/asterisk/keys/asterisk.crt
tlsprivatekey=/etc/asterisk/keys/asterisk.key
```

Configure PJSIP on `/etc/asterisk/pjsip.conf` as follows:

```
[webrtc_client]
type=aor
```

```
max_contacts=5
remove_existing=yes

[webrtc_client]
type=auth
auth_type=userpass
username=webrtc_client
password=webrtc_client ; This is a completely insecure password!  Do NOT expose this
                       ; system to the Internet without utilizing a better password.

[webrtc_client]
type=endpoint
aors=webrtc_client
auth=webrtc_client
dtls_auto_generate_cert=yes
webrtc=yes
; Setting webrtc=yes is a shortcut for setting the following options:
; use_avpf=yes
; media_encryption=dtls
; dtls_verify=fingerprint
; dtls_setup=actpass
; ice_support=yes
; media_use_received_transport=yes
; rtcp_mux=yes
context=default
disallow=all
allow=opus,ulaw
```

Additionally, for asterisk configurations, make a backup of the following files in /etc/asterisk/ directory and copy the same files from CRM directory /helpline/configs/ readily configured into the asterisk configuration directory /etc/asterisk/

```
manager.conf
pjsip.conf
extensions.conf
confbridge.conf
```

After the configurations, ensure you restart asterisk for the configurations to take effect.

AMI Service
This service provides a link between the CRM and asterisk and powers the telephony module on the CRM. Copy files provided separately as VoiceApps to /var/lib/

Create a service using systemd by copying the following to /etc/system/system/voiceapps.service

```
[Unit]
Description=Asterisk AMI Proxy
After=network.target mariadb.service

[Service]
Type=simple
#Restart=always
```

```
User=voiceapps
Group=voiceapps
WorkingDirectory=/var/lib/voiceapps/bin
ExecStart=/var/lib/voiceapps/bin/muu
#ExecStartPost=/var/lib/voiceapps/bin/ami.sh
[Install]
WantedBy=multi-user.target
```

System Access

The application requires that access be done through a secure link and this requires installation of an SSL access. With domain configuration, it becomes simpler. Within the local network, configure a domain with SSL and where the domain is for public access, follow the instructions below to add the domain to local host.

Open notepad as administrator on windows machine, trace the file “OS DRIVE/system32/drivers/etc/hosts” and add the following line

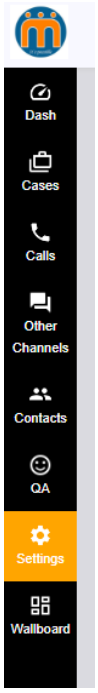
<<Local IP>> <<access domain>>
e.g

```
192.168.8.201 helpline.sematanzania.org
```

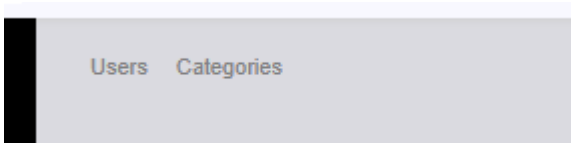
Save the file and you are ready to access the system on https://helpline.semtanzania.org/helpline/ on local network

System Configuration

This is an administrative function within the system which makes management easy. This is accessible through the Settings link on the system menu.



This provides two sections: Users and Categories.

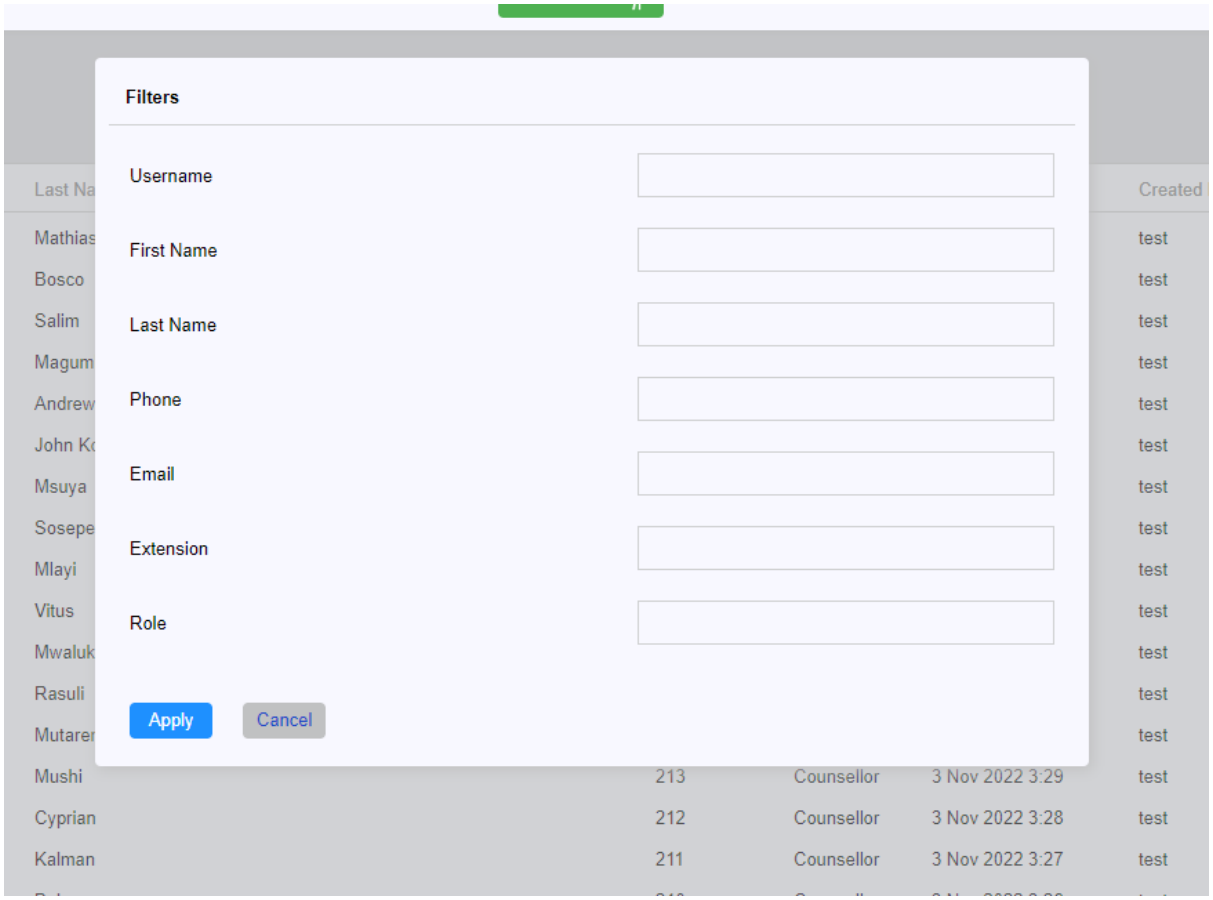


Users

This is a user management function, by default displays the list of created users which can be filtered based on user creation parameters. The function provides for two buttons on top right: Filter and New User.

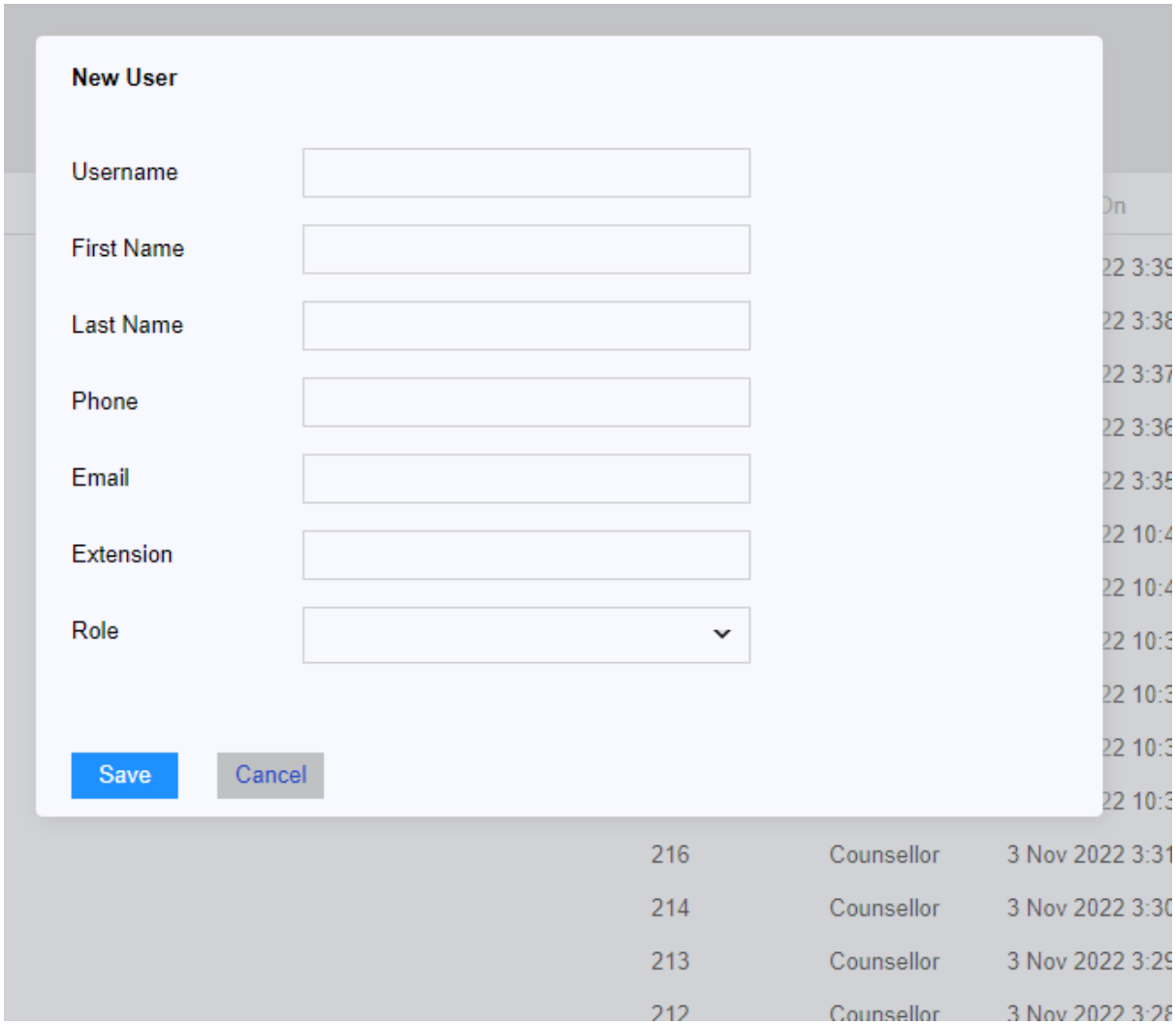
User Management function buttons

User Filter Form

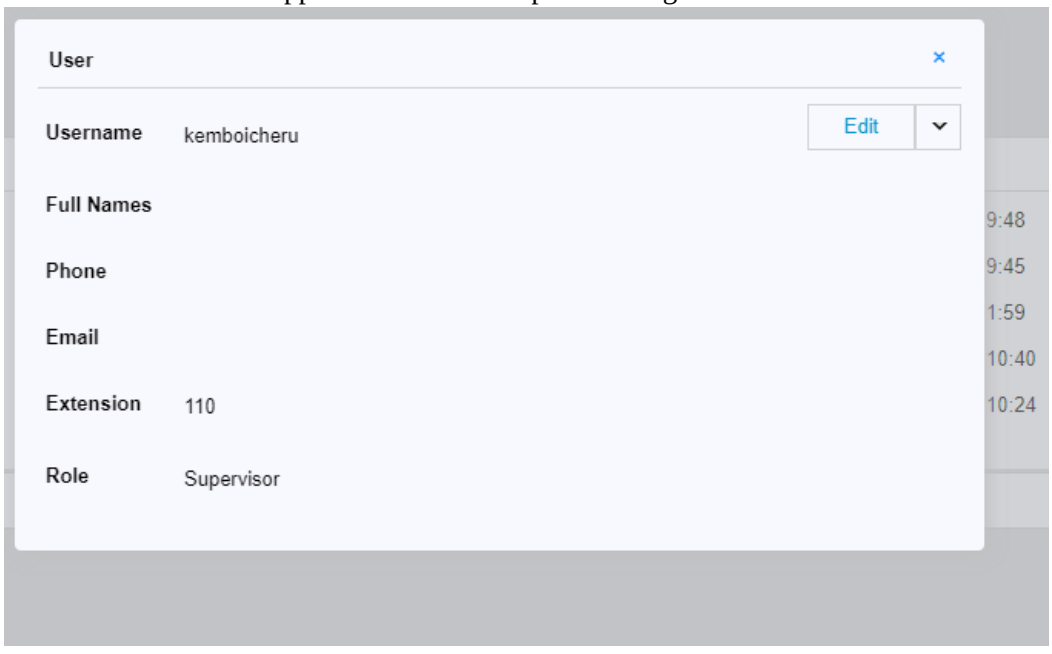


New User

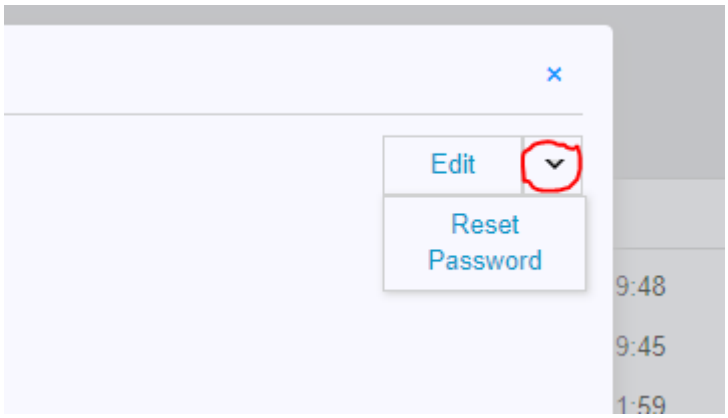
This button gives a form for user creation.



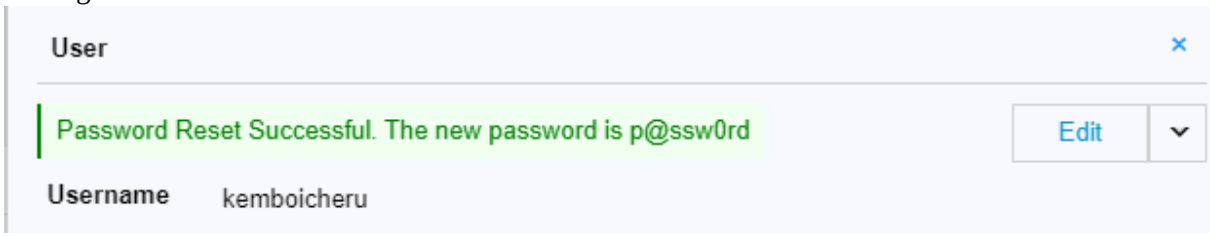
Once the user details are filled and submitted, a success message will appear and an option to edit the user. The same screen appears for user edit upon clicking a user on the list.



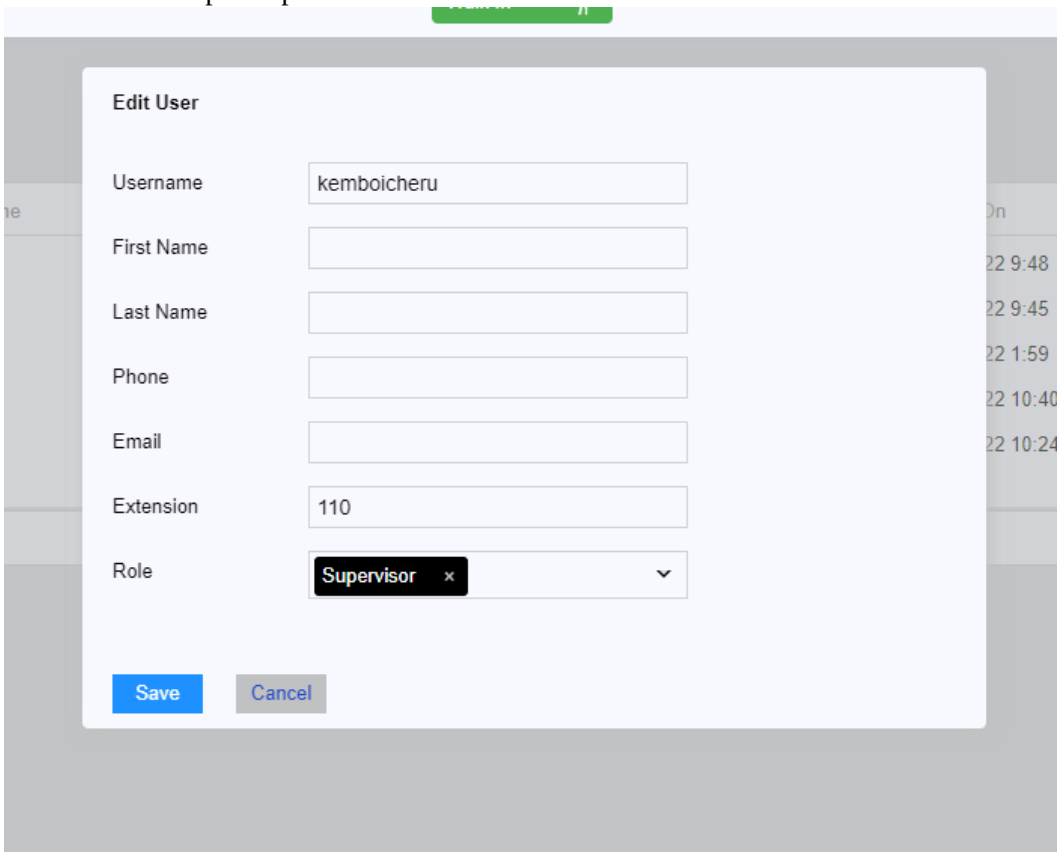
The angle arrow pointing downwards gives the administrator an option to reset user password to the default p@ssw0rd for any user.



Upon reset, a message and the default password is displayed if the reset was successful and an error message if not.

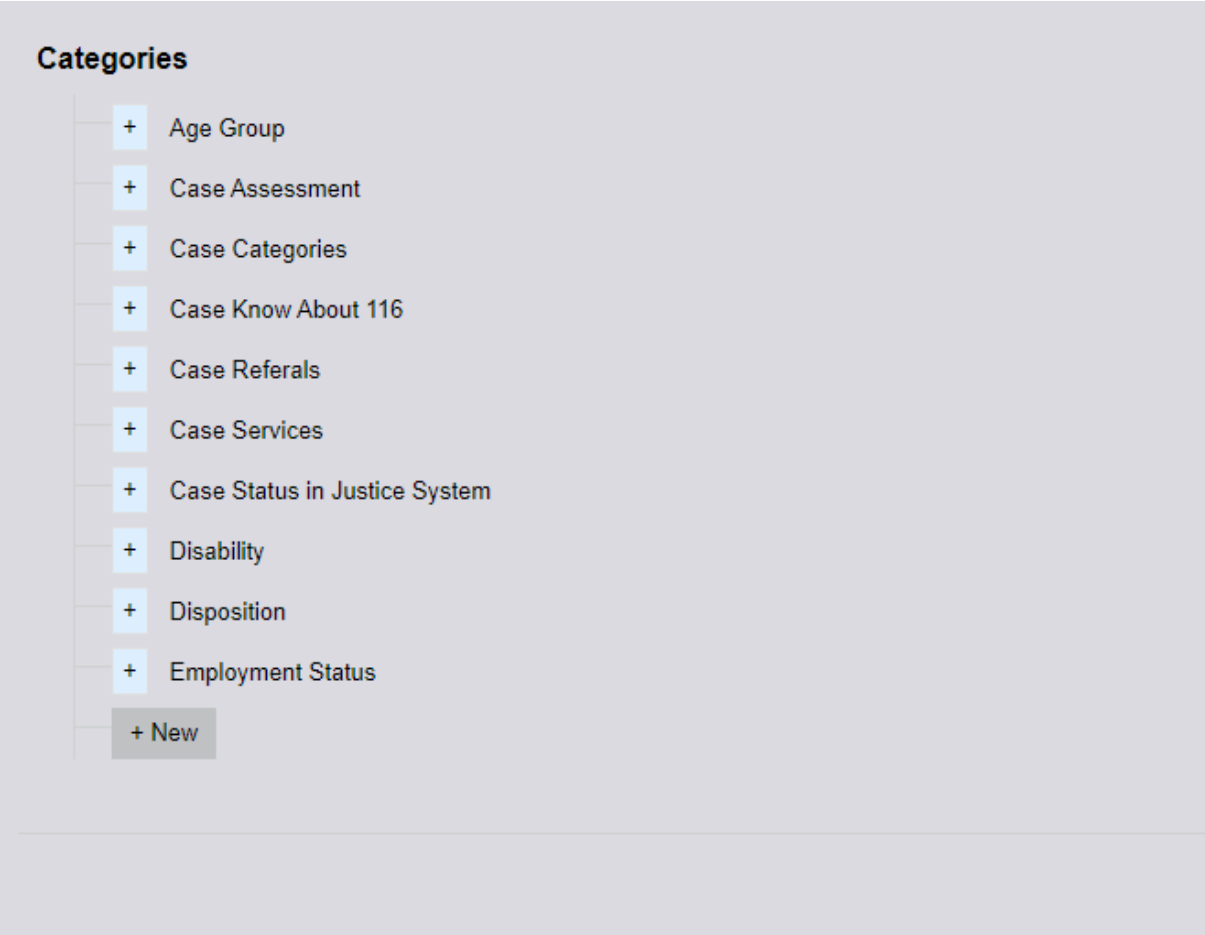


The edit button opens up a user edit form for the selected user as shown below:



Categories.

This is a tree view setting of all form field selectable options. An administrator can create or add to the existing options.



The different categories are mapped to the different and relevant form fields.

Conclusion

This technical document followed step by step shall end up in a working solution for a call center as provided by Bitz IT Consulting. It is good to keep in mind the dynamics of the open source community and updates in the installation procedures for upgraded versions of the dependencies where upgrades are not avoidable.

Technical Manual Sign-Off

<table>
<tr><td colspan="2">By signing this document, I acknowledge that I have received stated deliverables to the agreed quality levels.</td></tr>
<tr><td rowspan="2"></td><td>Signature:</td></tr>
<tr><td>Date:</td></tr>
<tr><td rowspan="2"></td><td>Signature:</td></tr>
<tr><td>Date:</td></tr>
<tr><td></td><td>Signature:</td></tr>
<tr><td></td><td>Date:</td></tr>
</table>