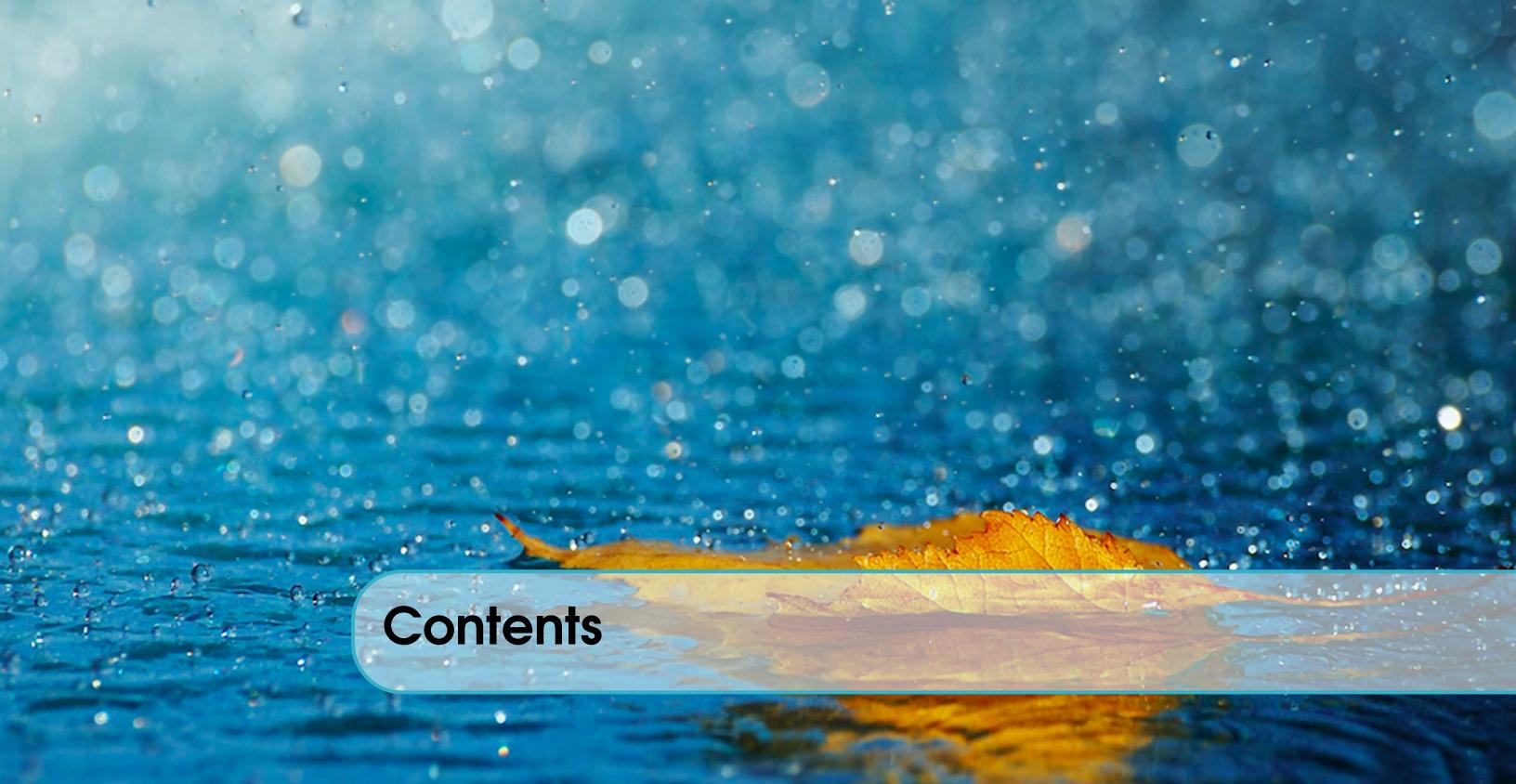


mTracPro Documentation

mTrac Powered by RapidPro

GoodCitizen Co. Ltd



Contents

1	Introduction	5
1.1	What is mTracPro?	5
1.2	About the Upgrade	5
1.2.1	Structure of the Codebase	5
1.3	mTracPro Context Diagram	7
2	mTracPro Components	9
2.1	RapidPro	9
2.2	Dispatcher-2	9
2.3	Web Interfaces	9
2.4	FSync	10
2.5	Reporter Synchronization	10
3	Installation	11
3.1	Prerequisites	11
3.2	Deployment	11
3.2.1	Deploying mTracPro	11
3.2.2	Deploying Dispatcher-2.1	13
3.3	Configurations	13
3.3.1	mTracPro Configurations	13
3.3.2	Dispatcher-2.1 Configurations	16

4	Web Interfaces Guide	17
4.1	Login	17
4.2	Reporters Management	18
4.2.1	Adding a Reporter	18
4.2.2	Editing or Deleting a Reporter	19
4.2.3	Filtering Reporters	20
4.3	Facilities	20
4.3.1	Report Completeness	20
4.3.2	Sending SMS	21
4.4	Reports Approval	21
4.5	Data Entry	22
4.6	Dispatcher-2.1 Web Interfaces	23
4.6.1	All Requests	24
4.6.2	Completed Requests	24
4.6.3	Ready Requests	24
4.6.4	Failed Requests	25
4.6.5	Search Requests	26
4.6.6	Interoperable Apps Settings	27
4.7	Administrative Interfaces	28
4.7.1	User Management	28
4.7.2	Groups Management	29
4.7.3	Facility Sync	30
4.7.4	Audit Log	31
4.7.5	SMS Log	31

1. Introduction

1.1 What is mTracPro?

mTracPro is a version of mTrac¹ that uses RapidPro as its SMS engine. It is built to have all the functionality of the previous production version which is built using RapidSMS. mTracPro comes with newer web interfaces that are responsive and work well on just any mobile/tablet device.

1.2 About the Upgrade

mTracPro has been build as an open source project and its source code has been made publicly available on GitHub under the following repository:

<https://github.com/sekiskylink/mtracpro>

1.2.1 Structure of the Codebase

Our mTracPro code base has been organized in a structure that make it easy for new developers to adopt it and hence simplify the process of troubleshooting. Figure 1.1 below show the directory structure for the mtracpro source-code directory.

¹mTrac is an SMS and web-based data collection platform developed to enable health facility workers and community health workers to submit routine weekly reports.

```
sam@idev:~/projects% tree mtrackpro/web
mtrackpro
|-- Makefile
|-- README.md
|-- development.txt
|-- doc
|-- licenses
|-- production.txt
|-- schema
|-- scripts
`-- web
    |-- __init__.py
    |-- app
    |   |-- __init__.py
    |   |-- bridge
    |   |-- controllers
    |   |-- models
    |   |-- tools
    |   |-- views
    |-- export_reporters.py
    |-- lib
    |   |-- __init__.py
    |-- load_reporters.py
    |-- local_settings.py
    |-- log
    |-- main.py
    |-- parse_kannel_log.py
    |-- reporters1.csv
    |-- send_scheduled.py
    |-- settings.py
    |-- static
    |   |-- css
    |   |-- downloads
    |   |-- font-awesome
    |   |-- fonts
    |   |-- help
    |   |-- images
    |   |-- img
    |   |-- js
    |-- sync_facilities.py
    |-- test
    |   |-- test.py
    |   |-- test_flow.py
    |-- upload_rapidpro_reporters.py
    `-- urls.py
```

Figure 1.1: mTracPro Source Code Directory Structure

The main directories include the following:

- `mtracpro/web` - has the settings files, other important directories like `app` and `static`
- `mtracpro/web/app` - has the directories where most of the code lies:
 - `app/controllers` - has the handlers for all pages and the code for the APIs. Each single page in the application has its own controller.
 - `app/views` - has all the HTML templates
- `mtracpro/web/static` - This holds all our static files including CSS, JavaScript and the images.

1.3 mTracPro Context Diagram

Figure 1.2 below shows the context diagram for mTracPro.

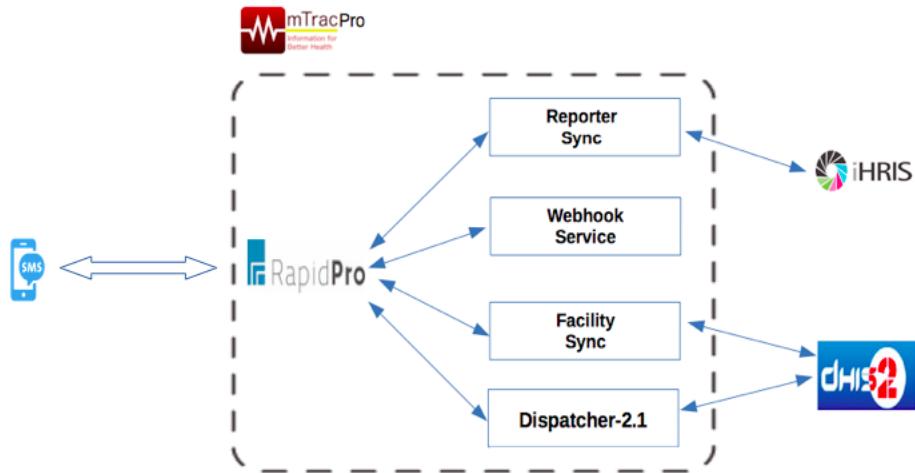


Figure 1.2: mTracPro Context Diagram



2. mTracPro Components

2.1 RapidPro

RapidPro is a hosted service for visually building interactive messaging applications.

RapidPro now has a couple of features that make it a suitable SMS engine to serve the needs of mTrac. Such features include:

- Contacts
- Flows
- Campaigns
- Triggers
- Channels

2.2 Dispatcher-2

Dispatcher-2 is data exchange middleware that is used for the exchange of data between two or more applications. It is primarily used to exchange data between mTrac and DHIS 2.

The latest version (dispatcher-2.1) has been improved to ensure that its use with mTracPro is a bit more smooth. Some of the new features include:

- JSON support - Now one can use the JSON data format while using the DHIS2 Web API to send data values from mTracPro to DHIS 2. Previously, only XML was supported
- Advanced Configuration - Now most advanced configuration options of the middleware can be managed through a configuration file.

2.3 Web Interfaces

mTracPro has been built with improved web interfaces that have been made responsive. The web interfaces replicate most of all the functionality of the old mTrac version. The web interfaces

also include management web interfaces for Dispatcher-2 and those for Facility Synchronization management.

2.4 FSync

FSync is an mTrac module that is responsible for Health Facility Synchronization with DHIS 2. It was originally developed as a separate application, but has now been integrated into mTracpro. It has been improved to include web interfaces to simplify the common synchronization tasks.

2.5 Reporter Synchronization

The Report Synchronization component of mTracPro helps to synchronize contacts from the mTracPro database to RapidPro. This has been designed to work with the Reporters from the old mTrac version, but also designed to be easy to integrate with iHRIS.



3. Installation

3.1 Prerequisites

The following are what you will need in order to deploy mTracPro:

- [RapidPro](#), Tool for visually building scalable interactive messaging applications.
- [PostgreSQL 9.4](#) or later
- [Kannel](#), an SMS/WAP Gateway - specifically gwlib
- [Nginx](#), an HTTP and reverse proxy server
- [Supervisor](#), a process control system
- [Web.py](#), a web framework for Python.

3.2 Deployment

The deployment instructions in this chapter are fairly biased towards Ubuntu.

3.2.1 Deploying mTracPro

Clone mTracPro

To clone mTracPro, issue the following commands on your terminal. Preferably in the /var/www directory on Ubuntu.

```
$ cd /var/www  
$ git clone https://github.com/sekiskylink/mtracpro.git
```

Build Virtual Environment

Using a virtual environment is our recommended way to run your mTracPro installation. The pip dependencies are in the `production.txt` file. Our default `WORKON_HOME` is `/var/www/virtualenvs`

```
$ cd mtracpro  
$ mkvirtualenv mtracpro  
(mtracpro)$ pip install -r production.txt
```

Create the mtrackpro Database

```
$ createdb -U postgres mtrackpro
```

Connect as superuser and import the mtracpro.sql script in the schema directory.

```
$ psql -U postgres mtrackpro
mtrackpro=# \i schema/mtracpro.sql
```

For the Uganda instance, a database dump with pre populated data has been created. This includes administrative units and reporters data.

Add mTracPro to Supervisor

We have created a sample process that we add to supervisor to start mTracpro using uwsgi. In our case, we have added the following snippet to the file /etc/supervisor/conf.d/mtracpro.conf

```
[program:mtracpro]
command=/var/www/virtualenvs/mtracpro/bin/uwsgi --socket 127.0.0.1:9090
          --wsgi-file main.py --pidfile /var/www/mtracpro/web/mtracpro.pid
stdout_logfile=/var/log/mtracpro/mtracpro.log
user=www-data
directory=/var/www/mtracpro/web
autostart=true
autorestart=true
redirect_stderr=true
stopsignal=QUIT
```

Deploy mTracPro under Nginx

To deploy mTracpro under Nginx, add the following snippet into a configuration file with in Nginx's conf.d directory. In our case it is /etc/nginx/conf.d/mtracpro.conf

```
server {
    listen 80;
    server_name mtracpro.gcinnovate.com;
    root /var/www/mtracpro/webhooks;
    access_log /var/log/nginx/mtracpro.access.log;
    error_log /var/log/nginx/mtracpro.error.log debug;
    location / {
        include uwsgi_params;
        uwsgi_pass 127.0.0.1:9090;

    }
    location /static{
        alias /var/www/mtracpro/web/static; # your project's static files
    }
}
```

3.2.2 Deploying Dispatcher-2.1

Clone Dispatcher-2.1

To clone Dispatcher-2.1, issue the following commands on your terminal.

```
$ git clone https://github.com/sekiskylink/dispatcher-2.1.git
```

Building and Installing Dispatcher-2.1

To build and install dispatcher-2.1 ensure that you have gcc, gwlib and essential C development libraries. Most of the dependencies can be revealed by the configure command. While in the directory where you cloned dispatcher-2.1, issue the following commands.

```
$ cd dispatcher-2.1
$ ./configure
$ make
$ sudo make install
```

Adding Dispatcher-2.1 Configuration File and Log Directory

While still in the dispatcher-2.1 directory, issue the following commands.

```
$ sudo mkdir -p /etc/dispatcher2
$ sudo cp docs/dispatcher2.conf /etc/dispatcher2
$ sudo mkdir -p /var/log/dispatcher2
```

Add Dispatcher2 to Supervisor

Dispatcher2 can be started as a supervisor process. Add the following snippet to your supervisor's conf.d directory. By default we add it to the file /etc/supervisor/conf.d/dispatcher2.conf

```
[program:dispatcher2]
command = /usr/local/bin/dispatcher2d -c /etc/dispatcher2/dispatcher2.conf
directory = /usr/local/bin
redirect_stderr = true
stdout_logfile=/var/log/dispatcher2/dispatcher2.log
stopsignal = INT
user = postgres
```

3.3 Configurations

3.3.1 mTracPro Configurations

The configurations/settings we talk about in this section are those found in the `setting.py` file under the `web` directory of the mtracpro codebase. These settings can be overridden by creating a `local_setting.py` file in the same directory.

General Configurations

Here we focus on the general configurations like hosts, usernames, passwords, and special URL endpoints. These are saved a Python dictionary called config within the settings file. The figure below shows the configurations.

Format:

```
config = {'var-name': 'value', ...}
```

Variable Name	Description	Default Value
db_name	Name of the mtracpro database	mtrackpro
db_host	The name or IP address of the machine running the database.	localhost
db_user	The username with which mtracpro will connect to the database.	postgres
db_passwd	The database user's password.	postgres
db_port	The database port	5432
logfile	The log file for mtracpro request	/tmp/mtrackpro-web.log
Dispatcher2 Specific		
dispatcher2_queue_url	The URL to call while queuing requests in dispatcher2	
dispatcher2_username	The recognized dispatcher2 username	
dispatcher2_password	The dispatcher2 user's password	
dispatcher2_certkey_file	A PEM encoded SSL certificate and private key file to be used with SSL server connections to dispatcher2	/etc/dispatcher2/ca-cert.pem
DHIS 2 Specific		
dhis2_user	The DHIS 2 username for synchronizing facilities and reports submission	
dhis2_passwd	The DHIS 2 user's password	
orgunits_url	The DHIS 2 organisationUnits URL	
sync_url	The URL for synchronizing (creating/updating) health facilities in mTracPro	
sync_user	The username used for accessing the sync service	
sync_passwd	The password for the sync user	
RapidPro Specific		
api_url	The RapidPro API URL	http://localhost:8000/api/v2/
api_token	The RapidPro API Token used for authentication	

Figure 3.1: mTracpro General Configurations

RapidPro Flows/Webhook Specific Configurations

Here we focus on those configurations used by the Webhooks called by RapidPro. The most important of these configurations is the MAPPINGS dictionary which holds the various flow variables (indicators), their description and the corresponding DHIS 2 Combo ID and DHIS 2 ID. These settings are very helpful when building the payload to submit to DHIS 2. Figure 3.2 show a sample of this configuration.

```

96 MAPPING = {
97     'apt_emtct_expected': {
98         'descr': 'Expected eMTCT Mothers on Appointment',
99         'dhis2_combo_id': 'gGhClrV5odI',
100        'dhis2_id': 'xVpHSWaIBEX'
101    },
102    'apt_emtct_missed': {
103        'descr': 'eMTCT Missed Appointments',
104        'dhis2_combo_id': 'gGhClrV5odI',
105        'dhis2_id': 'LYT11E96lC5'
106    },

```

Figure 3.2: Sample MAPPINGS Config

The other configurations used by the webhooks are specific to our CASES & DEATHS flows. For each of those flows, we want to be sure of the expected position of a particular indicator with in the SMS, the delimiter and the number of expected indicators. Figure 3.3 shows how we set these configurations.

```

22 CASES_POSITIONS = {
23     'ma': 0, 'dy': 1, 'sa': 2, 'af': 3, 'ae': 4, 'ab': 5, 'mg': 6, 'ch': 7, 'gw': 8,
24     'me': 9, 'nt': 10, 'vf': 11, 'pl': 12, 'tf': 13, 'yf': 14, 'tb': 15, 'md': 16, 'pd': 17
25 }
26 DELIMITER = '.'
27 KEYWORDS_DATA_LENGTH = {
28     'cases': 16,
29     'death': 18,
30 }
31

```

Figure 3.3: Indicator Position, Delimiter & No of. Indicators for CASES and DEATH Forms.

Dispatcher2 Related Configurations

Here we focus on the settings in the mTracpro settings file that relate to queuing requests into Dispatcher2.

```

410 PREFERRED_DHIS2_CONTENT_TYPE = 'json'
411 XML_TEMPLATE = """
412 <dataValueSet xmlns="http://dhis2.org/schema/dxf/2.0" dataSet="V1kJRs8CtW4"
413     completeDate="%completeDate)s" period="%period)s" orgUnit="%orgUnit)s">
414 <dataValues>
415     %(dataVales)s
416 </dataValues>
417 </dataValueSet>
418 """
419
420 DEFAULT_DATA_VALUES = {
421     'cases': "<dataValue dataElement='fc1vvNhzu7d' categoryOptionCombo='gGhClrV5odI' value='0' />",
422     'death': "<dataValue dataElement='YXIu5CW9LPR' categoryOptionCombo='gGhClrV5odI' value='0' />"
423 }

```

Figure 3.4: DHIS 2 XML Payload Template

3.3.2 Dispatcher-2.1 Configurations

Dispatcher2 has its own configuration file which follows the yaml style. The default path for the configuration file is /etc/dispatcher2/dispatcher2.conf. Figure 3.5 shows the configurations, their description and default values.

Format:

`var-name: value`

Variable Name	Description	Default Value
host	The name or IP address of the machine running the database.	localhost
database	Name of the database. The database need not exist as it will be created and populated automatically the first time dispatcher2 is started.	dispatcher2
user	The username with which the server will connect to the database.	postgres
password	The database user's password.	postgres
http-port	Port on which the dispatcher2 server should listen for commands from the user interface	
logdir	Directory where log files are written (file name dispatcher2.log)	/var/log
use-global-submission-period	Whether to use a global submission period for all apps using dispatcher2	true
start-submission-period	The hour marking the start of the global reports submission period	6
end-submission-period	The hour marking the end of the global reports submission period	22
max-retries	Maximum number of times a submission request will be retried before declared a failure.	3
max-concurrent	Maximum number of concurrent worker threads	5
default-queue-status	The status that will be assumed by requests inserted into dispatcher2's queue	ready
use-ssl	If set to true, HTTP transactions with the dispatcher2 server utilize the secure HTTP protocol (HTTPS). May be used to secure communications with the dispatcher2 server. Requires that the ssl parameters below are set.	false
ssl-client-certkey-file	A PEM encoded SSL certificate and private key file to be used with SSL client connections. This certificate is used for the HTTPS client side only, i.e. for URL requests to SSL-enabled HTTP servers.	
ssl-server-certkey-file	A PEM encoded SSL certificate and private key file to be used with SSL server connections. This certificate is used for the HTTPS server side only.	
ssl-trusted-ca-file		

Figure 3.5: Dispatcher2 Configuration Parameters

Request Filters

Submission ID:	<input type="text"/>		
Creation Date:	YYYY-MM-DD	to	YYYY-MM-DD
Source App:	Select Source		
Destination App:	Select Source		
Report Type:	Select Report Type		

Request Filters

Facility:	<input type="text"/>	
Status:	Select Status	
Body Format:	Select Content-Type	
Year:	Select Year	
Week:	Select Week	

4. Web Interfaces Guide

4.1 Login

The mTracPro web application can be accessed using the following URL:

<http://mtracpro.gcinnovate.com>

To login, provide a username and password using the Login interface.

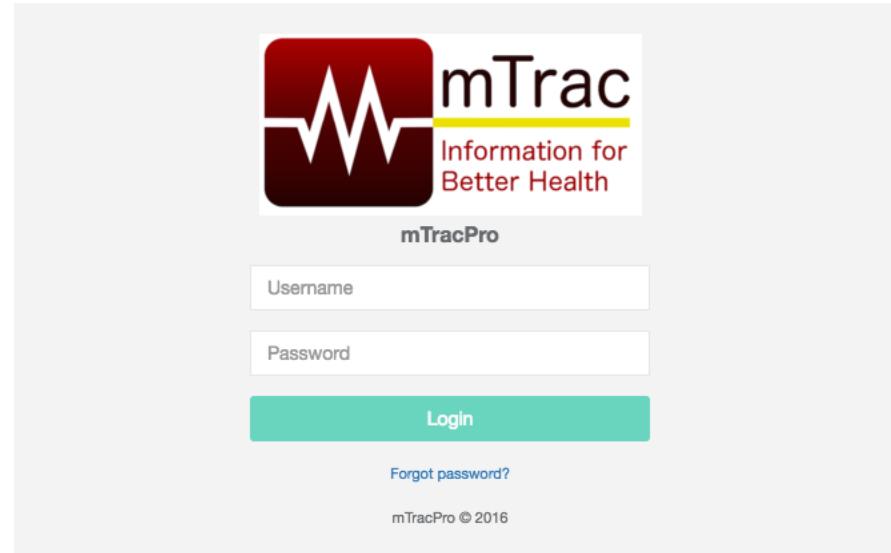


Figure 4.1: mTracPro Login Interface

Once logged in, you will have a set of links on the left navigation sidebar as shown in the figure below. The links displayed depend on your access rights.

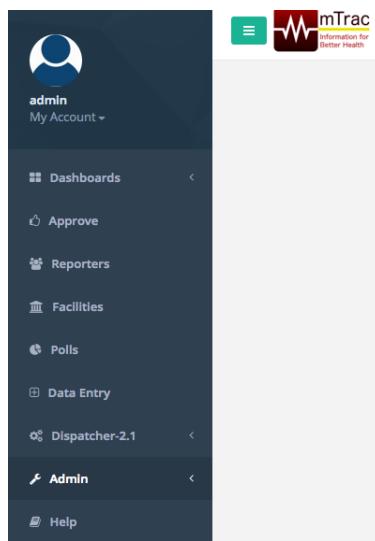


Figure 4.2: Left Navigation Sidebar

4.2 Reporters Management

To view the reporters page, click the “Reporters” link in the left navigation sidebar. Figure 4.3 below shows a sample reporters page.

Registered Reporters						
	Name	Telephone	Role(s)	District	Facility	Last Reported
<input type="checkbox"/>	Tom Opolot	256705579299	HC	Amuria	Abeko HC II NGO	

Figure 4.3: Reporters Page

4.2.1 Adding a Reporter

Click the “Add/Edit Reporter” tab at the top of the Reporters page. This will lead you to a validated form like that shown in Figure 4.4. Fill out this form and click the Save Reporter button to add the new reporter.

The screenshot shows the 'Add/Edit Reporter' form. It includes fields for District (Abim), Subcounty (Abim Town Council), Facility (Abim HOSPITAL), Role (HC selected), First Name (Faith), Last Name (Musawo), Telephone (+256), Other Tel (empty), Email (empty), and a Save Reporter button.

Figure 4.4: Add Reporter Form

4.2.2 Editing or Deleting a Reporter

The Reporters page shows a table listing reporters that are registered into the database. Each of these entries show action buttons that allow you to either edit or delete the reporter details. See Figure 4.5

The screenshot shows the 'Registered Reporters' listing page. It displays a table with columns: Name, Telephone, Role(s), District, Facility, Last Reported, and Action. The Action column contains edit and delete buttons. Red arrows point to the 'Edit Button' and the 'Delete Button'.

Name	Telephone	Role(s)	District	Facility	Last Reported	Action
Tom Opolot	256705579299	HC	Amuria	Abeko HC II NGO		
Okello Wonyima Isaac	256752762244	HC,FHD	Abim	Abim HOSPITAL		
Tsam Odongo	256752463793	HC	Apac	Abongomola HC III		
Akiror	256782504261	HC	Soroti	Agiriroi HC II		

Figure 4.5: Reporters Listing

Once the edit button is clicked, the “Add/Edit Reporter” form like that in Figure 4.4 above is populated with the details of the reporter. On the other hand, deleting a reporter prompts the user to confirm before deleting the record.

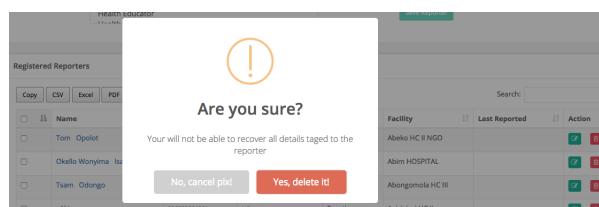


Figure 4.6: Reporter Delete Prompt

4.2.3 Filtering Reporters

To filter through the registered reporters, select the “Filter Reporters” tab on the Reporters page and type the search text, which could be either a name, or telephone number for the reporter. Clicking the “Search Reporter” button should return reporters matching the search text. See Figure 4.7

The screenshot shows a web-based application interface for managing reporters. At the top, there are two tabs: "Add/Edit Reporter" and "Filter Reporters". Below the tabs is a search bar labeled "Free Form Search:" containing the text "Samuel". To the right of the search bar is a green button labeled "Search Reporter". The main content area is titled "Registered Reporters" and contains a table with the following data:

	Name	Telephone	Role(s)	District	Facility	Last Reported	Action
<input type="checkbox"/>	Samuel Okabo	256782536729	HC	Nakapiripirit	407 Brigade HC III		

Figure 4.7: Reporters Filtering

4.3 Facilities

To view the Health Facilities in the system, click the “Facilities” link in the left navigation sidebar. This will open the Facilities page that shows a table listing the facilities. This will only show facilities within the district if the user account is a district account. Figure 4.8 below shows a sample facilities page.

The screenshot shows a web-based application interface for managing facilities. At the top, it displays the mTrac logo and the text "Logged in as Samuel Sekiwere" with a "Log out" link. The main content area is titled "Health Facilities" and contains a table with the following data:

	Name	Level	District	Last Reported	Has Complete Rpt	Action
1	Abim DHO Office	DHO's Office	Abim			
2	Abim HOSPITAL	General Hospital	Abim			
3	Abim Obolokome HC II	HC II	Abim			

Figure 4.8: Facilities Listing

4.3.1 Report Completeness

The facilities page also shows report completeness for each facility – under the column “**Has Complete Rpt**”. This column displays a Yes or No button which one can click for additional details about report completeness. This feature helps users to quickly point out which facilities haven’t submitted all the required reports. If all the mandatory reports have been sent, a green ‘Yes’ button is displayed, otherwise a red ‘No’ button is displayed.

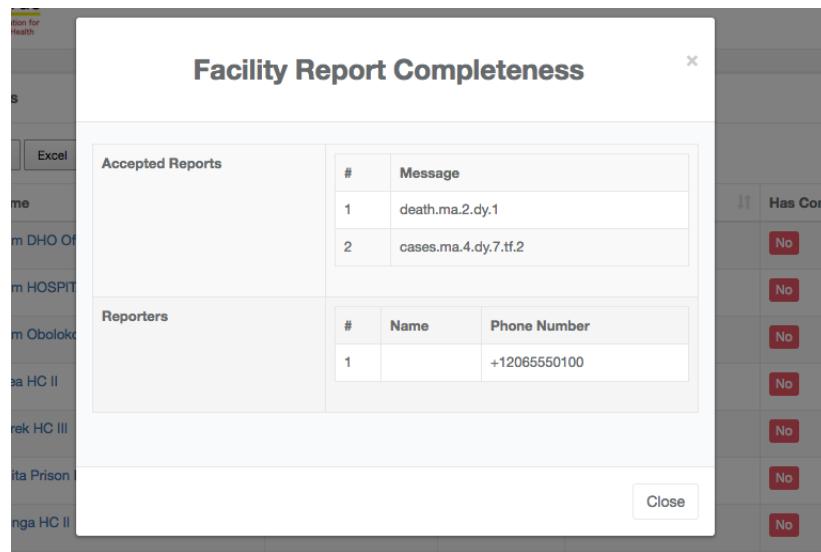


Figure 4.9: Report Completeness Pop-up

4.3.2 Sending SMS

You can send SMS to the reporters affiliated to a particular health facility. This can be done by clicking the envelope icon in the actions column of the facility record. This action leads to a Send SMS pop-up where you can specify the recipients group and type the message to send out.

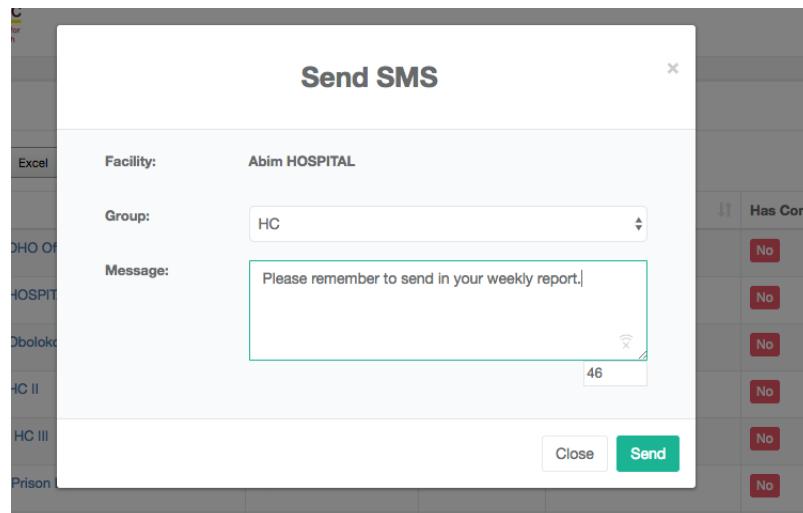
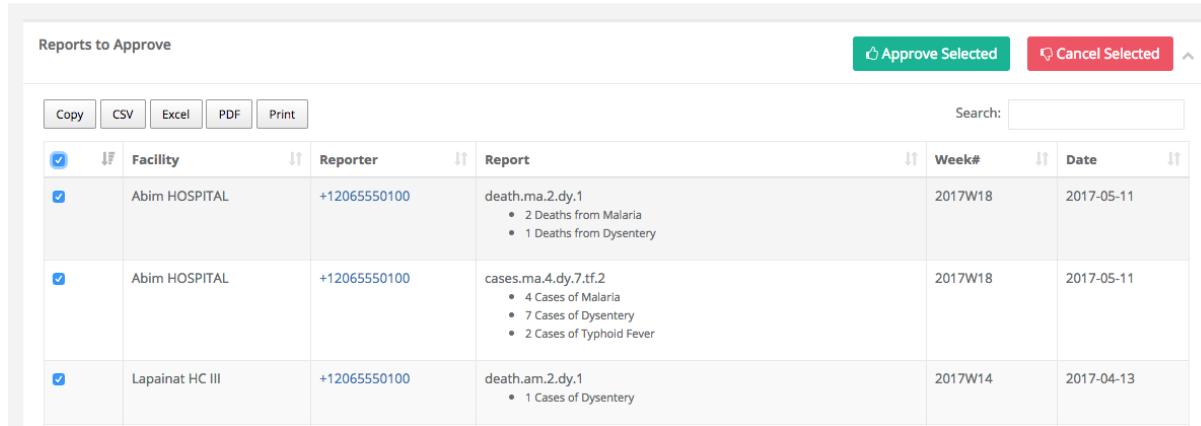


Figure 4.10: Facility Send SMS Pop-up

4.4 Reports Approval

To view or manage reports to approve, click the “Approve” link in the left navigation sidebar of your web page while logged in. This will show a table listing with the reports one can approve. Approving

a report will lead to immediately queuing it for submission to DHIS 2. Figure 4.11 below shows how this listing looks like.



The screenshot shows a table titled "Reports to Approve". At the top right are buttons for "Approve Selected" (green) and "Cancel Selected" (red). Below the table are buttons for "Copy", "CSV", "Excel", "PDF", and "Print". A search bar is also present. The table has columns: Facility, Reporter, Report, Week#, and Date. Three rows are listed:

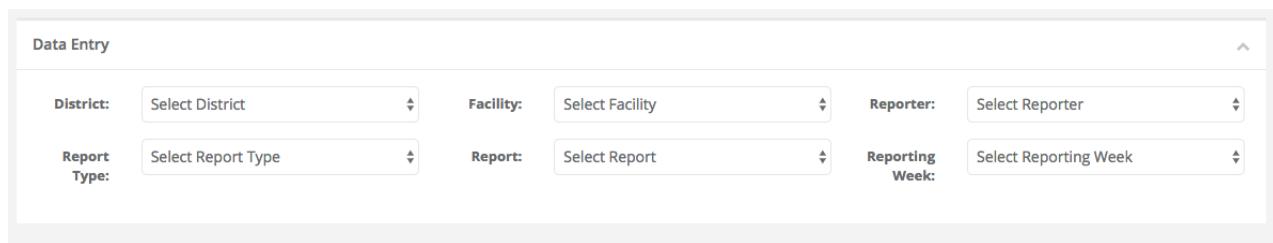
	Facility	Reporter	Report	Week#	Date
<input checked="" type="checkbox"/>	Abim HOSPITAL	+12065550100	death.ma.2.dy.1 • 2 Deaths from Malaria • 1 Deaths from Dysentery	2017W18	2017-05-11
<input checked="" type="checkbox"/>	Abim HOSPITAL	+12065550100	cases.ma.4.dy.7.tf.2 • 4 Cases of Malaria • 7 Cases of Dysentery • 2 Cases of Typhoid Fever	2017W18	2017-05-11
<input checked="" type="checkbox"/>	Lapainat HC III	+12065550100	death.am.2.dy.1 • 1 Cases of Dysentery	2017W14	2017-04-13

Figure 4.11: Reports to Approve

To approve a report for submission to DHIS 2, you simply tick the check-box in the first column of the reports listing and the hit the “Approve Selected” button. You can also choose to invalidate selected reports by clicking the “Cancel Selected” button at the top of the table.

4.5 Data Entry

The data entry functionality helps in the capture of late reports, or even those reports which are corrections of what could have been sent to the system. The Data Entry page can be accessed by clicking the “Data Entry” link in the left navigation sidebar of your web page while logged in. Figure 4.12 below shows what the initial data entry form looks like.



The screenshot shows a form titled "Data Entry". It contains six dropdown menus:

- District: Select District
- Facility: Select Facility
- Reporter: Select Reporter
- Report Type: Select Report Type
- Report: Select Report
- Reporting Week: Select Reporting Week

Figure 4.12: Data Entry Form

To capture a report, you select the district, health facility, the reporter, report type, the specific report, and its reporting week. Once the report is selected, a form with all indicators for that particular report is displayed. Figure 4.13 below shows an example of a report selected to be filled out.

Saving the captured report will eventually queue the report for submission to DHIS2.

The screenshot shows a 'Data Entry' form. At the top, there are dropdown menus for 'District' (Abim), 'Facility' (Alerek HC III), 'Reporter' (Achayo Irene (256779617125)), 'Report Type' (HMIS 033B Report), 'Report' (ARV Report, highlighted in green), and 'Reporting Week' (2017W18). Below these are two input fields: 'HIV screening test kits' and 'Nevirapine Therapy'. A green 'Save Report' button is at the bottom.

Figure 4.13: Sample Report Selection for Data Entry

4.6 Dispatcher-2.1 Web Interfaces

The Dispatcher-2.1 web interfaces have been integrated into the mTracpro web interfaces. In order to access them, please click the “Dispatcher-2.1” dropdown in the left navigation sidebar. Figure 4.14 below shows what the expanded dropdown is like.

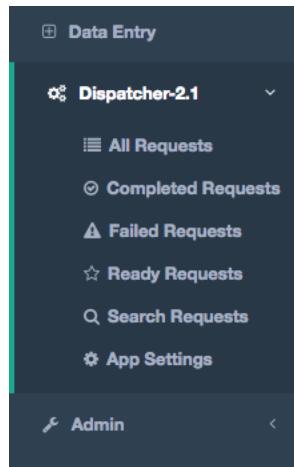


Figure 4.14: Dispatcher-2.1 Dropdown

All the web interfaces for Dispatcher-2.1 show a listing of reports that were queued for submission to DHIS 2 with their respective status. The following are the different values and meanings for the status supported by Dispatcher-2.1:

- **pending** - This is the status for reports pending approval for submission to DHIS 2
- **ready** - Any approved report is marked with status ready and can be directly pushed to DHIS2
- **completed** - This is the status for a successfully submitted report.
- **failed** - Any report for which submission to DHIS 2 fails is marked with status failed. The failure could range from network connectivity to errors in submission.
- **canceled** - Reports marked with status canceled can either be disapproved reports or those we do not intend to submit to DHIS 2 at all.

4.6.1 All Requests

The “All Requests” page shows a listing all the reports in the queue irrespective of their status.

The screenshot shows a table titled "Requests" with the following data:

Facility	Reporter	Report	Week	Creation Date	Status	Action
Abim HOSPITAL	+12065550100	death.ma.2.dy.1	2017W18	2017-05-11	canceled	<input type="button" value="Details"/> <input type="button" value="Delete"/>
Abim HOSPITAL	+12065550100	cases.ma.4.dy.7.tf.2	2017W18	2017-05-11	ready	<input type="button" value="Details"/> <input type="button" value="Delete"/>
Lapainat HC III	+12065550100	death.am.2.dy.1	2017W14	2017-04-13	ready	<input type="button" value="Details"/> <input type="button" value="Delete"/>
Lapainat HC III	+12065550100	cases.ma.6.tf.3	2017W14	2017-04-13	pending	<input type="button" value="Details"/> <input type="button" value="Delete"/>

Figure 4.15: All Requests

4.6.2 Completed Requests

The “Completed Requests” page shows a listing of all reports with status completed. Figure 4.16 below show what this listing is like. You can choose to retry already completed reports by selecting each report you want to retry and the hit the “Retry Selected” button.

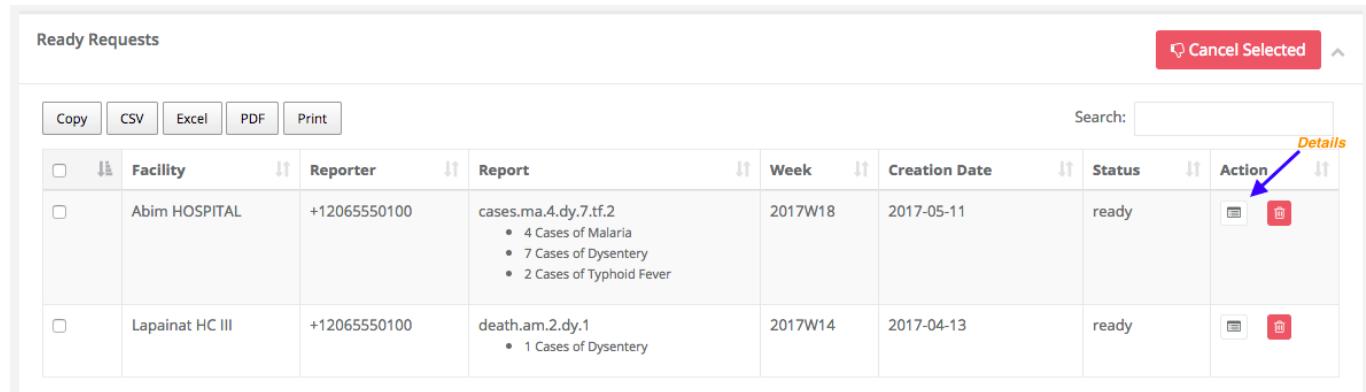
The screenshot shows a table titled "Completed Requests" with the following data:

Facility	Reporter	Report	Week	Creation Date	Status	Action
Abim DHO Office	+12065550100	cases.tf.1.ab.1.dy.2.mg.0.ch.1	2017W13	2017-04-02	completed	<input type="button" value="Details"/> <input type="button" value="Delete"/>
Lapainat HC III	+12065550100	cases.ma.3.dy.6	2017W13	2017-04-02	completed	<input type="button" value="Details"/> <input type="button" value="Delete"/>

Figure 4.16: Completed Requests

4.6.3 Ready Requests

The “Ready Requests” page shows a list of reports that have been approved and ready for dispatcher-2.1 to submit them to DHIS 2. You can choose to cancel any of the ready requests by selecting each of the reports you want to cancel and it shall be marked canceled and not submitted to DHIS 2 in turn. Each entry of the ready reports has a details button that can be used to show the extra details of the request. These extra details also show a well formatted JSON body of the request. This makes it easy for one troubleshooting requests submission.

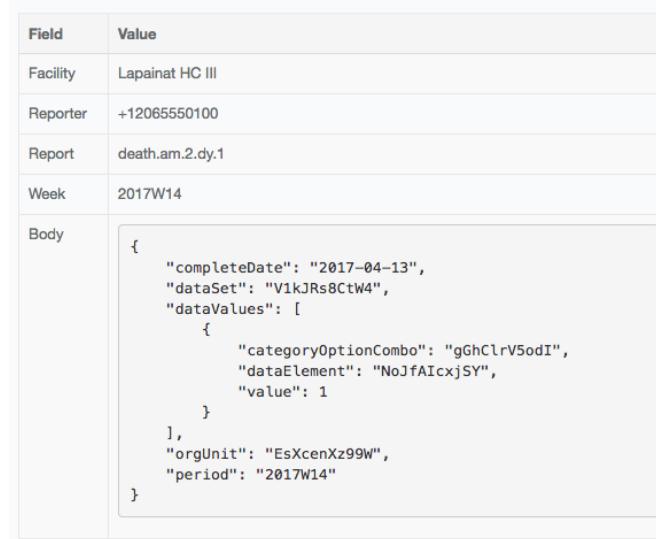


The screenshot shows a table titled "Ready Requests". The columns are: Facility, Reporter, Report, Week, Creation Date, Status, and Action. There are two rows of data. The first row corresponds to "Abim HOSPITAL" with reporter "+12065550100" and report "cases.ma.4.dy.7.tf.2" which includes a list of three items: "4 Cases of Malaria", "7 Cases of Dysentery", and "2 Cases of Typhoid Fever". The second row corresponds to "Lapainat HC III" with reporter "+12065550100" and report "death.am.2.dy.1" which includes a list of one item: "1 Cases of Dysentery". The "Action" column contains two buttons: a blue "Details" button and a red "Delete" button, with a blue arrow pointing to the "Details" button.

Ready Requests							<input type="button" value="Cancel Selected"/>
	<input type="button" value="Copy"/>	<input type="button" value="CSV"/>	<input type="button" value="Excel"/>	<input type="button" value="PDF"/>	<input type="button" value="Print"/>	Search:	
	Facility	Reporter	Report	Week	Creation Date	Status	Action
<input type="checkbox"/>	Abim HOSPITAL	+12065550100	cases.ma.4.dy.7.tf.2 • 4 Cases of Malaria • 7 Cases of Dysentery • 2 Cases of Typhoid Fever	2017W18	2017-05-11	ready	<input type="button" value="Details"/> <input type="button" value="Delete"/>
<input type="checkbox"/>	Lapainat HC III	+12065550100	death.am.2.dy.1 • 1 Cases of Dysentery	2017W14	2017-04-13	ready	<input type="button" value="Details"/> <input type="button" value="Delete"/>

Figure 4.17: Ready Requests

Figure 4.18 below shows a sample Popup with a JSON formatted request.



The screenshot shows a "Request Details" popup. It has a table with columns "Field" and "Value". The fields listed are Facility, Reporter, Report, Week, and Body. The "Body" field contains a JSON object:

```

{
  "completeDate": "2017-04-13",
  "dataSet": "V1kJRs8CtW4",
  "dataValues": [
    {
      "categoryOptionCombo": "gGhClrV5odI",
      "dataElement": "NoJfAIcxjSY",
      "value": 1
    }
  ],
  "orgUnit": "EsXcenXz99W",
  "period": "2017W14"
}

```

Figure 4.18: Requests Details Popup

4.6.4 Failed Requests

When requests are queued for submission, some may fail to be successfully submitted. This failure can be as result of a couple reasons such as; network connectivity and errors returned from destination server (DHIS 2 Web API). The “Failed Requests” page shows a table listing of the failed requests. Additional information concerning the reason for the failure is provided in the details popup. Figure 4.19 shows a sample failed requests page.

Failed Requests							C Retry Selected
	Facility	Report	Reporter	Week	Creation Date	Status	Action
<input type="checkbox"/>	EsXcenXz99W	+12065550100	cases.ma.3.dy.6 • 3 Cases of Malaria • 6 Cases of Dysentery	2017W13	2017-04-02	failed	
<input type="checkbox"/>	NREoMszwQZW	+12065550100	cases.ma.10.ab.1.ch.1.tf.1 • 10 Cases of Malaria • 1 Cases of Animal Bites • 1 Cases of Cholera • 1 Cases of Typhoid Fever	2017W14	2017-03-30	failed	

Figure 4.19: Failed Requests

You can retry failed requests by selecting the ones you want to retry and then click the “Retry Selected” button at the top right of the failed requests table.

4.6.5 Search Requests

Searching through queued requests, that is; the failed, ready completed and the canceled comes in handy when troubleshooting reports submission between mTrac and DHIS 2. The “Search Requests” page has a form with many fields that you can use to search through the queued requests. Searching using the form will display requests matching the search criteria dictated by the search fields. Figure 4.20 shows the requests search form with the associated field.

Request Filters	
Submission ID:	<input type="text"/>
Creation Date:	<input type="text"/> YYYY-MM-DD to <input type="text"/> YYYY-MM-DD
Source App:	<input type="text"/> Select Source
Destination App:	<input type="text"/> Select Source
Report Type:	<input type="text"/> Select Report Type
Reporter Tel:	<input type="text"/>
Request Filters	
Facility:	<input type="text"/>
Status:	<input type="text"/> Select Status
Body Format:	<input type="text"/> Select Content-Type
Year:	<input type="text"/> Select Year
Week:	<input type="text"/> Select Week
Search	

Figure 4.20: Requests Search Form

4.6.6 Interoperable Apps Settings

The “App Settings” page gives you the ability to define the settings for the applications you want to connect using Dispatcher2. These settings include: the name of the application, username and password used to connect to it, the authentication method supported by the application, and others. Figure 4.21 shows a form on the “App Settings” used for capturing the various settings/configurations of an application.

Figure 4.21: Application Settings Form

The “App Settings” page also displays a table with all the applications supported by Dispatcher2 (see Figure 4.22). You can edit or delete an application by clicking the relevant action button in the last column of the table.

#	Name	Username	Password	Allowed Apps/Servers	Auth Method	Action
4	dhis2	tester	foobar	• mtrackpro • mtrack	Basic Auth	
1	localhost	tester	foobar	• mtrack	Basic Auth	
3	mtrack	tester	foobar	• mtrackpro	Basic Auth	

Figure 4.22: Supported Application List

4.7 Administrative Interfaces

4.7.1 User Management

The “Users” page helps you to manage system users. The management of users includes; creating, editing and deleting the user accounts.

Adding Users

To add, a user, fill in the users form that appears at the top of the “Users” page then click the “Save User” button. Figure 4.23 shows a form used to create a user.

Figure 4.23: New User Form

Editing/Deleting Users

To delete a user, click the red delete button in the action column of the table listing the users. A prompt to confirm deletion will then show. A user will then be deleted up on confirmation. Figure 4.24 shows a list of users with the edit and delete action buttons.

#	Name	Telephone	Username	Email	Role	Action
1	Abim District		abim		District User	
2	Adjumani District		adjumani		District User	
3	Agago District		agago		District User	
4	Alebtong District		alebtong		District User	

Figure 4.24: Users List

4.7.2 Groups Management

The “Groups” page helps you with managing the groups/roles. This management involves creating the groups with their associated permissions for the various system modules, editing and deleting groups.

Adding Groups

To add, a group, fill in the Group details form that appears at the top of the “Groups” page, including the permissions on each module for that group and then click the “Save Group” button. Figure 4.25 shows a form used to create a group with its associated permissions.

Group Details	
Name:	<input type="text"/>
Description:	<input type="text"/>

Group Permissions	
Approve Reports:	<input type="checkbox"/> Read <input type="checkbox"/> Modify
Reporter Management:	<input type="checkbox"/> Read <input type="checkbox"/> Modify
Polls Management:	<input type="checkbox"/> Read <input type="checkbox"/> Modify
Reports:	<input type="checkbox"/> Read

Save Group

Figure 4.25: New Group From

Editing/Deleting Groups

To delete a group, click the red delete button in the action column of the table listing the groups. A prompt to confirm deletion will then show. A group will then be deleted upon confirmation. Figure 4.26 shows a list of groups with the edit and delete action buttons.

To edit a group and its associated permissions, click the green edit button. This will populate the group details form with the details of the form being edited.

#	Name	Description	Action
1	Partners	For the partners	
2	District User	For the district users	
3	Administrator	For the Administrators	

Figure 4.26: List of Groups

4.7.3 Facility Sync

The “**Facility Sync**” page helps to automate the process of synchronizing health facilities with DHIS2. When synchronizing health facilities with DHIS 2, we keep a local cache of all the facilities in DHIS 2 that we then use to track changes with the facilities in mTracPro. The FSync component runs a cron job (scheduled task) that routinely tracks the changes in order to synchronize the health facilities.

The “**Facility Sync**” page shows a list of the cached copy of health facilities from DHIS 2. This listing also has action buttons for synchronizing and deleting a facility from the cache. You can also choose to synchronize a select few facilities or perform a full sync using the buttons at the top right of the table listing facilities in the local cache. See figure 4.27

Health Facilities						
	District	Sub County	Facility Name	Level	DHIS2 ID	Action
<input type="checkbox"/>	Abim	Abim	Amita Prison HC II	HC II	Owwv3PblrZ2	
<input type="checkbox"/>	Abim	Abim	Atunga HC II	HC II	Ug8m4dtpeYZ	
<input type="checkbox"/>	Abim	Abim	Kanu HC II	HC II	ZotVvjcyDdV	

Figure 4.27: Form for Synchronizing with DHIS2 IDs

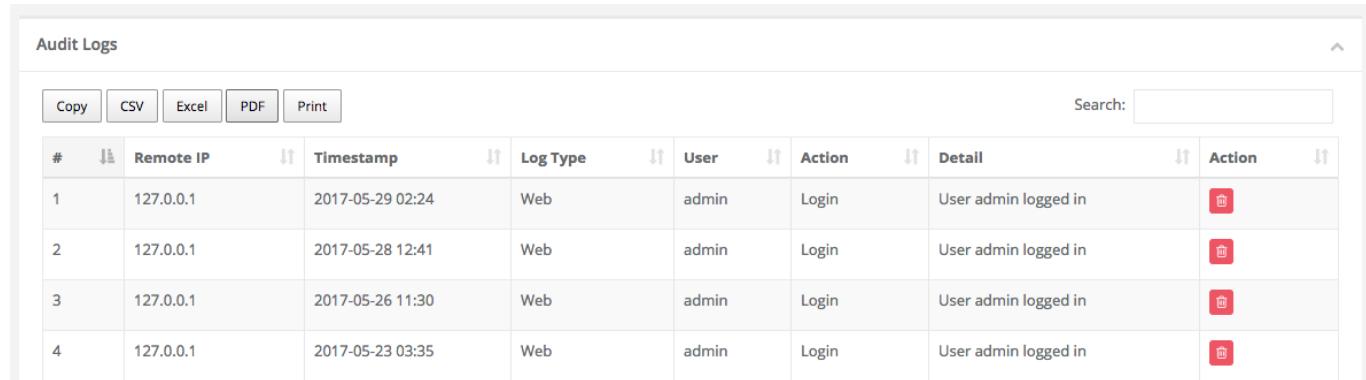
Synchronizing Health Facilities with a DHIS 2 ID

Each health facility has a unique DHIS 2 ID. We can use the facility ID or a list of IDs for health facilities in order to synchronize it/them with mTracPro. This can be done with the help of a form at the top of the page, where we input the list of comma separated IDs and click the “Sync Facilities” button. See figure 4.28

Figure 4.28: Form for Synchronizing with DHIS2 IDs

4.7.4 Audit Log

We have created means to audit and log the activity happening while using mTracPro. An administrative web interface has been created to help keep track of the activity happening through the use of the system. Some of the activities tracked include; login, logout, report approvals, bulks SMS sending, etc. Figure 4.29 below shows a sample “AuditLog” page.



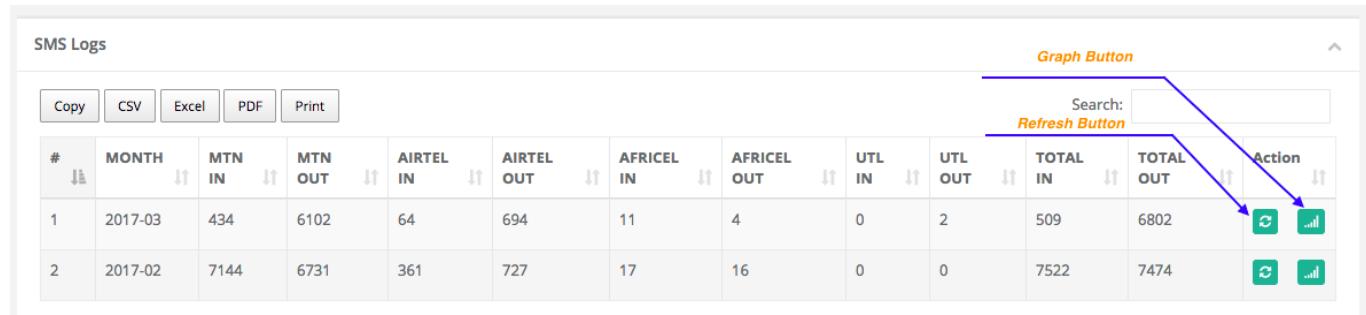
The screenshot shows a table titled "Audit Logs" with the following columns: #, Remote IP, Timestamp, Log Type, User, Action, and Detail. Each row contains a delete icon in the Action column. The data in the table is as follows:

#	Remote IP	Timestamp	Log Type	User	Action	Detail	Action
1	127.0.0.1	2017-05-29 02:24	Web	admin	Login	User admin logged in	
2	127.0.0.1	2017-05-28 12:41	Web	admin	Login	User admin logged in	
3	127.0.0.1	2017-05-26 11:30	Web	admin	Login	User admin logged in	
4	127.0.0.1	2017-05-23 03:35	Web	admin	Login	User admin logged in	

Figure 4.29: Requests Search Form

4.7.5 SMS Log

The “SMS Log” page shows a summary total of both incoming and outgoing message aggregated by mobile network and month. These totals are generated from the Kannel SMS logs and so they are very close approximations of the actual total value. This is true, especially when the script computing the total is run after short intervals (say after every 5 minutes - which is the default). Figure 4.30 shows a sample SMS Log listing.



The screenshot shows a table titled "SMS Logs" with the following columns: #, MONTH, MTN IN, MTN OUT, AIRTEL IN, AIRTEL OUT, AFRICEL IN, AFRICEL OUT, UTL IN, UTL OUT, TOTAL IN, TOTAL OUT, and Action. The Action column contains two buttons: a refresh button and a bar chart button. The data in the table is as follows:

#	MONTH	MTN IN	MTN OUT	AIRTEL IN	AIRTEL OUT	AFRICEL IN	AFRICEL OUT	UTL IN	UTL OUT	TOTAL IN	TOTAL OUT	Action
1	2017-03	434	6102	64	694	11	4	0	2	509	6802	
2	2017-02	7144	6731	361	727	17	16	0	0	7522	7474	

Figure 4.30: SMS Log

Each entry in the SMS Logs table has two action buttons in the Action column. The first is the refresh button which is used reprocess the logs and compute the current totals at the time of the refresh. The second button generates a bar chart showing both incoming and outgoing totals for each mobile network for the month. See figure 4.31

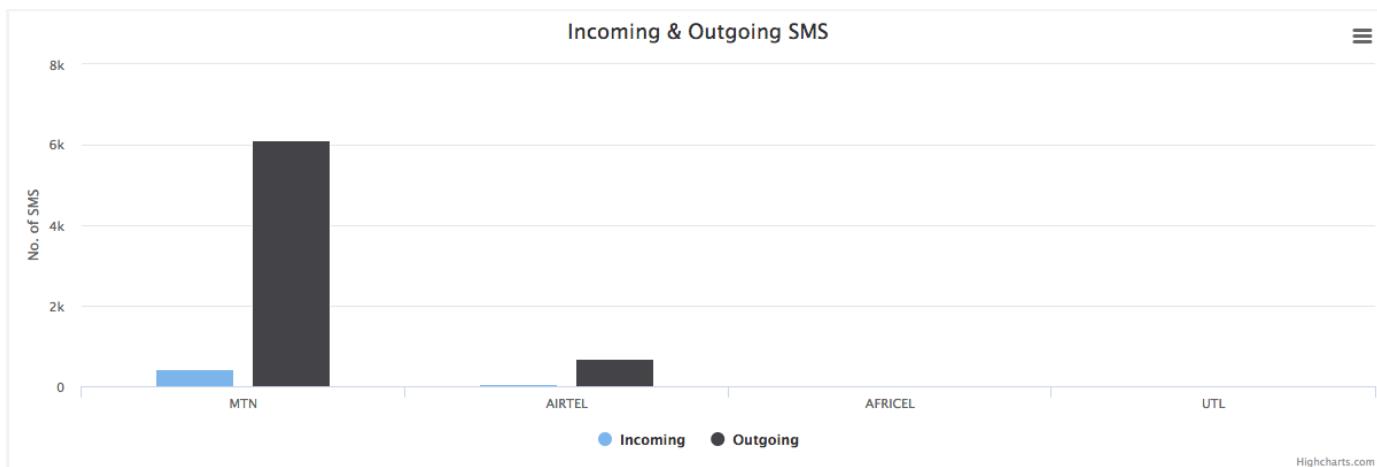


Figure 4.31: SMS Log Graph