

SAHANA for Italy

EVAcuees ASSistance - EVASS

Table of Contents

TABLE OF CONTENTS	2
INTRODUCTION	4
WHAT IS SAHANA EVASS.....	5
SCENARIOS.....	6
INSTALLATION	7
INSTALLING ON WINDOWS SYSTEMS.....	7
INSTALLING ON LINUX SYSTEMS	7
SECURITY	9
A. ORGANIZATION	9
B. DOMAIN / REALM.....	11
C. ROLE.....	12
OFFICES AND FACILITIES.....	19
SHELTERS.....	20
MANUAL REGISTRATION.....	21
IMPORT A CSV FILE.....	21
HOUSING UNITS	23
EVACUEES REGISTRATION INSIDE A SPECIFIC HOUSING UNIT	23
EVACUEES	24
EVACUEES REGISTRATION	24
MANUAL REGISTRATION	24
REGISTRATION THROUGH A CSV FILE	25
MAPPING / GEO LOCALIZATION.....	27
HOW TO CONFIGURE THE MAP	28
CONFIGURATION	30
CONFIGURING MODULES/000_CONFIG.PY	30
GMAIL CONFIGURATION:.....	31
HOTMAIL/LIVE CONFIGURATION:	31
CONFIGURING PRIVATE/TEMPLATES/EVASS/CONFIG.PY	31
MAP SETTINGS	32
STARTING EDEN.....	33
LOGIN.....	34
USERS REGISTRATION	36
SYSTEM ADMINISTRATOR REGISTERS A NEW USER	36
REGISTRATION THROUGH A FILE	36
REPORTING	38
USING EVASS - TIPS.....	39
USING MULTIPLE INSTANCES OF EVASS.....	39
COMPONENT'S DEPENDENCIES	40
LIST OF COMPONENTS.....	42
COMPONENT HIGH-LEVEL DESIGN.....	42
DATABASE CHANGES / EXTENSIONS	44

APPENDIX A - DELIVERABLES	46
APPENDIX B - EXAMPLE OF PRIVATE/TEMPLATES/EVASS/CONFIG.PY	47
APPENDIX C - DOCUMENT REFERENCES	58

Introduction

Sahana Eden is an Open Source Humanitarian Platform that can be used to provide solutions for the Disaster Management, Development, and Environmental Management sectors. The software is used worldwide and can be customized to meet the needs of every country or organization that chooses to adopt it. Starting from Sahana Eden it is possible to create a customized template to address some specific needs; this creates an extension of the standard and basic functions available in Sahana.

What is Sahana EVASS

This document describes a customization created in Sahana Eden to address specific needs of Italy, which is exposed to natural disasters like earthquakes and floods. The main goal of this project, internally named “Sahana for Italy”, is to support the organizations that work to help people before, during and after the emergency has occurred. A template of Sahana Eden called “**EVASS**” has been created to adapt the software to the needs of the Italian organizations, which act as stakeholders and partners for this project.

This document describes the result of the effort of a team of volunteers who developed this template in the context of a pro-bono project sponsored by Fondazione IBM Italia. The template EVASS has become part of the Sahana platform and is available to be downloaded by any country having the same requirements as Italy.

EVASS has developed totally new elements and modified some of the existing ones. Both new and modified elements have been created as modular pieces and are integrated into the main software. Sahana Eden includes many components developed over the years to take care of different aspects of an emergency (for a complete list, please refer to: [Sahana Eden Features.](#))

Sahana for Italy is the result of collaboration with one specific Italian Organization that identified a list of typical scenarios that they typically manage during the emergencies of recent years. These scenarios have been mapped into the existing Sahana and this has permitted the functional gap to be identified and new functionalities to be defined and added to the existing platform.

Scenarios

During an emergency there are many tasks to be run and Sahana includes the functions to execute some of them. “Sahana for Italy” has identified a macro area totally new and not yet addressed by Sahana platform: it’s about the **assistance to the population** during the disaster. To ensure a complete management of the “Evacuees”, EVASS is linked to the other existing Sahana functional areas normally used for managing daily tasks or other aspects of an emergency. Many functions to manage Evacuees are totally new developed modules; for scenarios involving instead administrative or daily tasks, we extended or modified the existing components in agreement with Sahana Foundation. The list below is the set of scenarios designed for Italy and developed in the context of the EVASS project but applicable also to other countries having the same priorities or needs:

What an organization adopting EVASS can do:

1. Run general administrative tasks to install, configure, and define authorized staff; making possible the use of the tool in accordance with the Italian data privacy regulation (very restrictive but applicable also to other European countries).
2. Register an event / emergency and the associated incidents.
3. Define and assign one or more organizations to manage the emergency. Define the Organization’s hierarchy. After this step define also the list of employees belonging to an Organization and authorized to work on that specific event, specifying their role and profile to grant the correct level of authorization.
4. Create new Shelters or associate existing ones to a specific event. Shelters provide a set of services and capabilities and can be organized as a subset of Housing Units.
5. Register evacuees and provide a temporary accommodation, keep track of their needs and ensure the best accomodation. Keep family or groups together and use a basic booking system to allow a transfer among the shelters.
6. Easily identify on a map the area of the emergency, shelters or other Points of Interest (Pol) in the proximity (Geo-Localization).
7. Provide tools to ensure a good and quick notification to the staff driving the emergency.

Scenario 5 requires a complete new development module.

Shelters (Scenario 4), required extensive changes in order to be linked to the evacuees (Scenario 5) and get benefits from it.

Other minor changes have been applied to other modules (already present in Sahana) to correctly implement the remaining scenarios.

EVASS has extended the list of configurable fields in order to grant backward compatibility, but, at the same time, we make it possible to link these to the new Evacuees module in order to implement the described scenarios.

Installation

The complete Installation procedure is available on the official wiki:

<http://eden.sahanafoundation.org/wiki/InstallationGuidelines>

When visiting the official repository of Sahana Eden code, there are two different versions: the frozen version and the version under development. The latter includes also EVASS and all the related extensions applicable to Italian organizations.

Sahana Eden supports different operating systems such as Windows, Linux and Mac. For a complete list, refer to the online documentation available on the [Sahana Wiki site](#).

Installing On Windows systems

EVASS and the entire Sahana platform can run on any Windows platform³. To install the application on a Windows system, create a suitable environment.

1. Download and install all python prerequisite libraries
<http://eden.sahanafoundation.org/wiki/InstallationGuidelines/Windows/Developer/Manual>
2. Download and install the Git client <http://msysgit.github.io/>
3. Create a new folder to contain all the Sahana Eden Software
4. Using the Git Shell, browse to the new folder
5. Using the Git Shell, download the web2py framework by running this command
`git clone https://github.com/web2py/web2py.git`
6. Using the Git Shell, download the Sahana Eden software by running these commands
`cd web2py`
`cd Applications`
`git clone https://github.com/flavour/eden.git`

Installing On Linux systems

EVASS and the entire Sahana platform can run on any Linux platform⁵. Currently Debian and Ubuntu are supported platforms. To install the application on Linux (Debian-Based Systems), create a suitable environment.

1. Download and install all python prerequisite libraries
<http://eden.sahanafoundation.org/wiki/InstallationGuidelines/Linux/Developer/Manual>
2. Download and install the Git client by running these commands
`sudo su - apt-get install git-core`
3. Create a new folder to contain all the Sahana Eden Software
4. Using the Ubuntu Terminal, browse to the new folder
5. Using the Ubuntu Terminal, download the web2py framework by running the command:

⁵ During the project, EVASS has been tested on Windows 7 and Ubuntu Server 12.04 and Ubuntu Desktop 14.04

git clone <https://github.com/web2py/web2py.git>

6. Using the Ubuntu Terminal, download the Sahana Eden software by running the commands:

cd web2py

cd Applications

git clone <https://github.com/flavour/eden.git>

Security

Sahana Eden makes available a security model to protect access to the data and resources to only pre-authorized users; it can be modeled to work with different levels of organizations. EVASS has adopted the same security model, but the implementation takes into consideration the operating model of the Italian organization where the complexity of layers is increased by the need to protect and segregate personal data from sensitive data.

So security in EVASS provides the possibility to limit the access to the resources and to the data to users NOT authorized or partially authorized.

When the SW Administrators register the user candidates to use the SW, they configure also the correct level of authorization to access the data and functions.

The authorization to access the data and functions is implemented by assigning to the new users two variables: realm/domain and role. The new users will only be allowed to view the data and functions defined by the administrator.

a. Organization

EVASS grants a rich and complex management of the organizations (delivered by the module org).

Organization information is grouped into different sets; the table below shows those currently active in EVASS:

Basic Details	Including the general information of an Organization, such as: name, type, web site, phone number (primary and secondary), and so on.
Branches	To Create / Update / Remove information about inherited branches of the root organization.
Offices	To define the list of offices belonging to the organization.
Facilities	To define the facilities belonging to the organization.
Staff and Volunteers	To define Staff and Volunteers belonging to the organization.
Roles	To define the role to assign to a registered user who will log in to the application and use it (see below, domain and role section).

The correct definition of the Organization's structure directly affects the correct utilization of the SW; it affects/denies access to the resources and data to the SW users.

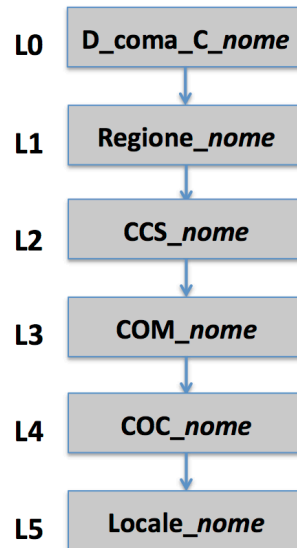
An Organization can be organized into multiple levels, each level can include multiple branches and each branch can be:

- Associated to Staff and Volunteers
- Associated to Shelters and Warehouses
- Assigned to Events and Incidents to be managed
- Associated with the evacuees requiring assistance

The Italian Organization (for which this version of the SW was initially created), manages the emergency with a well-defined hierarchical organization and this model is applied to manage all

the emergencies: it's the Italian model.

To standardize the approach and reduce human error, a standard template has been created to easily reproduce the organization's structure.⁴ This is its representation:



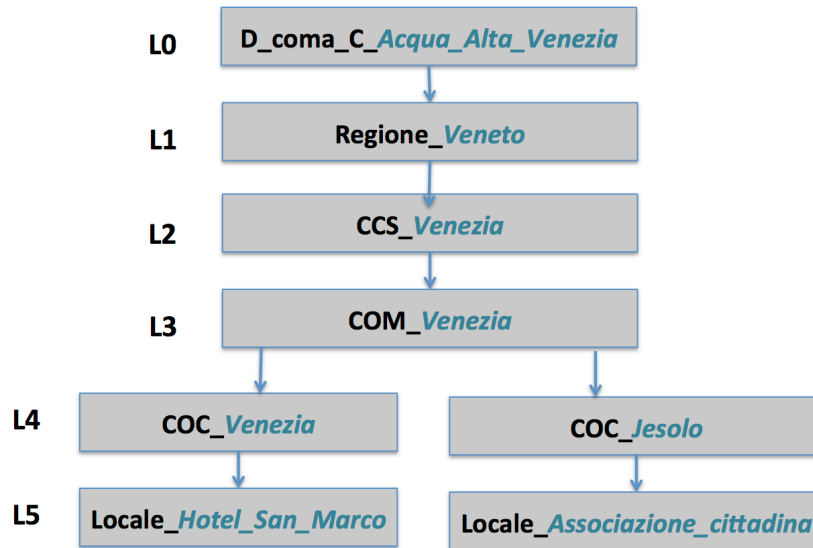
Each emergency to be managed requires the description of the entire organization and the assignment of the resources starting from the lowest level to the highest level depending on the required geo-political competencies.

Using the same organization's structure promises a better analysis from everyone both during the emergency and in the later phases.

The organization represented by the template is the minimum vertical extension; in the future it can be extended "horizontally" adding new branches depending on the goals of the organization.

⁴ The word "nome" present on each layer of the organization's template must be replaced with valid identifiers (that typically is linked with the specific organization involved in the event's management).

Es:



The definition of the organization in the SW can be done by:

a. Manual Definition of the organization

In the Organization menu, create L0 of the organization by submitting the command “Create Organization”. Move to the tab “Branches” of the created organization, and continue to create all the layers and their branches, selecting the command “Add” (from the contextual menu or clicking the right button).

b. Automatic Definition of the organization through IMPORT

The described template is represented in a CSV file available in the folder: */private/template/EVASS/organisation.csv*. To import this organization, from the Organization Tab, select “Import” and then choose the file. Then change the name of the branches (right-click and then select “Edit”).

Note: In the current version, a user can create new branches but cannot delete them. It is possible, however, to delete the entire organization.

In general, all the resources defined in EVASS should be associated to an organization in order to correctly define the operating environment.

When a group starts to use EVASS as the tool for their daily activities, it’s suggested that they create a default organization to which to define all the resources (not assigned to any specific events); it is considered a sort of container.

When a specific event is created and assigned to a specific organization, the resources can be defined as new ones or simply re-assigned to a specific event by picking them up from the default container created and populated at start time.

b. Domain / Realm

Each user must have a Domain / Realm assigned to it, defining the group of data that the user

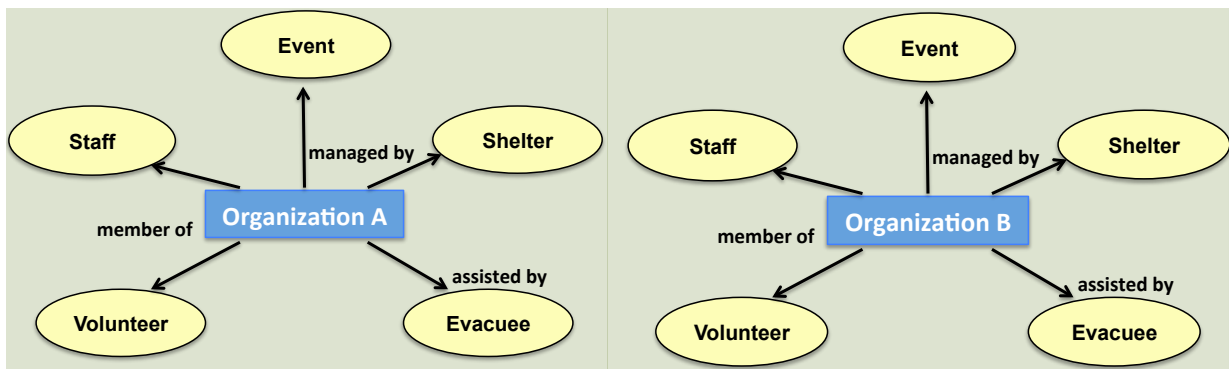
can access. To create correct access to data, it's suggested to create more than one Domain / Realm of data so that the end user with the correct profile will see only part of the information stored in the database.

In EVASS the **Organization** is considered to be key information when defining the Domain / Realm. Let's explain why by looking at some examples:

- A member of the staff is registered as part of an Organization
- A shelter is managed by an Organization
- An evacuee is assisted by a specific Organization

All this data is part of one specific Organization and is defined in the same domain.

A user authorized to use the SW is registered as part of an Organization and can access only the data available in the same domain.



Example:

Assuming that in EVASS there are two Organizations defined: **Organization A** and **Organization B**, each of them has registered staff and its members and volunteers.

A new user is registered and is assigned to Organization A: he can access and manage all the data defined in Organization A but cannot view data defined in Organization B.

It is possible to define a Domain / Realm named "All Entities", but when a user is assigned to it, he can access and view all the data belonging to all the Organizations defined in the SW.

c. Role

A role defines the set of actions that each user can run on the data available in the SW. Each user must be assigned at least one role; because Sahana EDEN organizes the data into database tables, a user's role defines how he can work on the table. In general the actions permitted for each row of the table are: Read/ Update/ Delete/ Edit.

Because all the modules and functions available in Sahana Eden are based directly on the database tables, when a user is assigned a role to manage or not manage tables, he indirectly is enabled or disabled to use the associated modules.

Only the SW Administrator can create new roles or modify existing ones:

[Administration](#) → [User](#) → [User Management](#) → [List all roles](#)

Each user can be assigned one or more roles.

Example: the SW Administrator assigns to user_A the roles of Staff Admin and Notification Sender. User_A will be able to access the modules Staff, Volunteers, Evacuees, and Shelters and can send notifications. He cannot access any other functions.

Following the suggestions of the Italian organization, EVASS already includes some pre-defined roles.

In EVASS some pre-defined roles are available (they have been defined according to the Italian organization specifications) even if the software Administrator can create new roles or modify existing ones by following the commands from the Administration menu:

[Administration](#) → [User](#) → [User Management](#) → [List all Roles](#).

Existing roles can be considered complementary and intuitive to better support the association User to Role and they can be aggregated into three macros:

- a. Base Roles or Configuration Roles: assigned to configure and manage the SW. In general only one role is assigned. The possible roles are:

Authenticated	Default role. Each registered user is automatically assigned this role. This role does not give access to any functions or resources: it only suggests the correct registration.
Administrator	FULL access to all the modules and functions. Preferably assigned to the Staff, Supervisor and members of the Command center
Map Admin	FULL access to the maps and their configurations. Preferably assigned to the Staff and members of the Command center

- b. Primary Roles: assigned to run some specific tasks. In general only one of the following roles is assigned.

Public Authority Reader	Access in READ to the modules of assistance: Evacuees and Shelter. Preferably assigned to a Supervisor user member of a local government unit.
Public Authority Admin	FULL access to the modules of assistance: Evacuees and Shelter. Preferably assigned to a user member of a local government unit with the authorization to register data.
Private User Reader	Access in READ to the modules of assistance: Evacuees and Shelter. Preferably assigned to a private user member of an organization dedicated to provide temporary assistance to the evacuees.

Private User Editor	FULL access to the modules of assistance: Evacuees and Shelter. Preferably assigned to a private user member of an organization dedicated to provide temporary assistance to the evacuees.
Volunteer Reader	Access in READ to the modules of assistance: Evacuees and Shelter. Access in READ to the modules to manage human resources: Staff and Volunteers. Preferably assigned to a volunteer providing assistance to the evacuees belonging to a well-defined shelter.
Volunteer Admin	FULL access to the modules of assistance: Evacuees and Shelter. Access in READ to the modules to manage human resources: Staff and Volunteers. Preferably assigned to a volunteer (Expert or benefit from the trust) providing assistance to the evacuees belonging to a well-defined shelter.
Staff Reader	Access in READ to the modules Staff, Volunteers, Evacuees and Shelters. Preferably assigned to Staff managing one or more shelters
Staff Admin	FULL access to the modules of Staff, Volunteers, Evacuees and Shelters. Preferably assigned to Staff managing one or more shelters.

- c. Secondary or complementary Role: assigned to run some specific tasks. In general more than one of the following roles is assigned and must be associated to one of the Primary Roles and/or Base Role:

Event Reader	Access in READ only to the Events and Incidents already registered. Preferably assigned to Staff directors or Staff with assigned responsibility.
Event Admin	FULL access only to the Events and Incidents already registered. Preferably assigned to Staff directors or Staff with assigned responsibility.
Organisation Reader	Access in READ only to the Organizations and related branches. Preferably assigned to Staff directors or Staff with assigned responsibility.
Organisation Admin	FULL access only to the Organizations and related branches.

	Preferably assigned to Staff directors or Staff with assigned responsibility.
Notification Sender	FULL access only to the module Messaging to send/receive notifications (Email/Sms/Twitter). Preferably assigned to Staff directors or Staff with assigned responsibility.
Medical Details Admin	FULL access only and exclusively to the medical data of the registered evacuees. Assigned to Medical Staff
Map Location Reader	Access in READ to the Locations (city, Point Of Interest) already registered in the software. Preferably assigned to Staff
Map Location Admin	FULL access to the Locations (city, Point Of Interest) already registered in the software. Preferably assigned to Staff directors or Staff with assigned responsibility.

Table 1.1 includes all the roles already defined and activated in EVASS. For each role there is the list of permitted actions on the database tables.

All: Access in CRUD to everything

R: Access only in Read mode

No: No access, authenticated only

YES: Access to the map

	Events	Incidents	Organizations	Offices	Facilities	Shelters	Evacuees	Groups	Medical Details	Person Registry	Staff	Volunteers	Map	Map Configuration	Locations	Points of Interest	Notifications
Authenticated	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO
Administrator	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	YES	ALL	ALL	ALL	YES
Map Configuration Admin	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES	ALL	ALL	ALL	NO

Public Authority Reader	NO	NO	NO	NO	NO	R	R	R	NO	NO	NO	NO	YES	NO	NO	NO	NO
Public Authority Admin	NO	NO	NO	NO	NO	ALL	ALL	ALL	NO	NO	NO	NO	YES	NO	NO	NO	NO
Private User Reader	NO	NO	NO	NO	NO	R	R	R	NO	NO	NO	NO	YES	NO	NO	NO	NO
Private User Admin	NO	NO	NO	NO	NO	ALL	ALL	ALL	NO	NO	NO	NO	YES	NO	NO	NO	NO
Volunteer Reader		NO	NO	NO	NO	R	R	R	NO	NO	R	R	YES	NO	NO	NO	NO
Volunteer Admin	NO	NO	NO	NO	NO	ALL	ALL	ALL	NO	NO	R	R	YES	NO	NO	NO	NO
Staff Reader	NO	NO	NO	NO	NO	R	R	R	NO	NO	R	R	YES	NO	NO	NO	NO
Staff Admin	NO	NO	NO	NO	NO	ALL	ALL	ALL	NO	NO	ALL	ALL	YES	NO	NO	NO	NO
Event Reader	R	R	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO
Event Admin	ALL	ALL	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO
Organization Reader	NO	NO	R	R	R	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO
Organization Admin	NO	NO	ALL	ALL	ALL	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO
Medical Details Admin	NO	NO	NO	NO	NO	NO	NO	NO	ALL	NO	NO	NO	YES	NO	NO	NO	NO
Notification Sender	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	YES
Map Location Reader	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	R	R	NO

Map Location Admin	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	ALL	ALL	NO
--------------------	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	----	-----	-----	----

Table 1.1

The next table 1.2 describes the roles and when to assign them.

Authenticated	Default role. Each user registered into Sahana Eden, automatically inherits this role. This role does not give access to any data, it only means that the user is correctly registered into the SW.
Public Authority Reader	Access in “READ only” mode to modules Evacuees and Shelter. When to assign: supervisor user member of a local government office.
Public Authority Editor	FULL access to modules Evacuees e Shelter. When to assign: supervisor user member of local government office with the rights to register data
Private User Reader	Access in “READ only” mode to modules Evacuees and Shelter. When to assign: private user member of a Shelter / Organization and committed to provide assistance to some evacuees.
Private User Editor	FULL access to modules Evacuees and Shelter. When to assign: private user member of a shelter / Organization and committed to provide assistance to some evacuees
Volunteer Reader	Access in “READ only” mode to modules Evacuees and Shelter. When to assign: volunteer who provides assistance to some evacuees inside a well-defined group of shelters.
Volunteer Editor	FULL access to modules Evacuees and Shelter. When to assign: volunteer (trusted person) volunteer who provides assistance to some evacuees inside a well-defined group of shelters.
Staff Reader	Access in “READ only” mode to modules Staff, Volunteer, Evacuees and Shelter. When to assign: Staff assigned to manage one or more shelters.
Staff Admin	FULL access to modules Staff, Volunteer, Evacuees and Shelter. When to assign: Staff assigned to manage one or more shelters.
Medical Details Admin	Complementary role: FULL access but only to medical data of registered evacuees. It must be assigned with other roles to grant access to the other types of data. When to assign: Medical Staff.
Organization Reader	Complementary role: Access in “READ only” mode to Organization data and its branches When to assign: Staff Directors
Organization Admin	Complementary role: FULL access mode to Organization data and its branches. When to assign: Staff Directors.
Notification Sender	Complementary role: FULL access mode only to module messaging to send/receive notifications (Email/SMS/Twitter). When to assign: Staff Directors.
Map Admin	Complementary role: FULL access mode only to the maps and their configurations When to assign: Staff Directors.

Administrator	FULL access mode to all the modules and functions. When to assign: Staff Directors.
---------------	--

Table 1.2

OFFICES and FACILITIES

Each Organization can be associated to Offices (Administrative and Operational) and Facilities (Warehouse, distribution centers, and so on).

For a practical and simplified management, Offices and Facilities can be represented on the map (GIS module). To enable their visualization on the map, during their registration it is important to provide the correct location positioning them in the Italian territory (“Find on Map”).

Offices and Facilities are defined in the Organization module (ORG), so the access to this information strictly depends on the rights to access the module Organization (see the paragraph [Roles](#)).

To create a new office, move to the Organization Tab:

Organizations-> Offices -> Create

To create a new facility:

Organizations-> Facilities -> Create

SHELTERS

The EVASS template enables the management of shelters in all the situations (before, during or after the emergency); there are two modules delivering all the functions needed to provide assistance to the population and they are the “shelters” (cr) and the “evacuees” (evr). The Shelter module organizes the information into many different sheets. Currently the active sheets are:

Basic Details	General information about the shelter, such as name, type, services, owner organization, contacts, and so on.
Status Report	Basic reports about the status and population in the shelter at a well-defined moment.
People Reservation	Reservation for a number of seats in a shelter for the future arrival of new evacuees.
People Registration	People registration in the shelter.
Staff	Staff currently assigned to manage the shelter.
Assign Staff	Assign a new member to the staff managing the shelter.
Housing Units	Description of the Housing Units of the shelters.
Send notification	Send notifications.

From the SW there is no difference among the different types of shelters: they are managed in the same way. A shelter with tents or hotels or a temporary facility is the same. Each shelter can be carefully described by the available services (presence of toilet, cafeteria, sewers, infirmary) and by the characteristics of the environment where the shelter is installed (streets, paved or dirt road, close to the sea, beach).

A shelter is defined by a set of housing units and the total capacity of a shelter is the sum of the capacity of each single housing unit. When a new housing unit is registered, the total capacity of the associated shelter is automatically incremented.

When a new evacuee is registered in a specific housing unit (and therefore also into a shelter), the available capacity of the shelter is updated as well as the total size of the population.

The shelters are also represented on the map in order to quickly view their distribution and status. This requires a complete registration of their position in the Italian territory (look for “Find on Map” button, in the tab “Basic details”).

To the shelters is applied the same security model described previously in order to protect data and information of the human persons registered there. So each shelter must be associated to an organization responsible for their management. The association shelter – organization establishes the integration of the shelter into the same domain of the organization. This is a fundamental and mandatory step to ensure the correct management of the information and to limit access to the data to unauthorized people.

Shelters not associated to any Organization: it is possible and in this case the access to the data is permitted only to the Administrator. This case happens at the beginning, when there is an initial phase of pre- population of the SW with the available resources (and independently from the events). To register a shelter, there are two methods: manually or by importing a CSV file.

Manual Registration

The SW Administrator or any user authorized to register new shelters (according to their roles), can manually create a new shelter and insert the data by using the appropriate sheet in the user interface. The registration includes two steps: during the first step, generic data is inserted into the sheet “basic details”; during the second step, additional details are registered in the additional sheets.

Import a CSV file

The CSV utilization is the method to automatically register more than one shelter with only one operation. The import action is available inside the module “shelter”; the shelter created automatically with the import, can be modified manually at a later stage. The CSV file must follow a template available in the software and it can be downloaded using the link “download template” (in the shelter sheet). The table below represents the fields in the CSV and what they represent.

Organization*	Optional	The shelter managing the event
Branch	Optional	The inherited organization branches
Name	Required	Name of the shelter
Country	Optional	Country in which the shelter is located
L1**	Optional	Region
L2**	Optional	Municipality
L3**	Optional	City
Address	Optional	Address of the shelter
PostCode	Optional	Postal Code
Lat	Optional	Latitude
Lon	Optional	Longitude
Capacity	Optional	Capacity (during the day)
Population	Optional	Population
Status	Optional	Status (Open / Closed)

* The name of the organization must correspond to the name of an already existing organization and previously registered (it is case insensitive). If the name does not exist, a new organization is created.

** The location fields must be completely compiled; the SW discards the template automatically when only one field is provided (example L3). This is an example of a CSV file:

```
Organisation,Branch,Name,Country,L1,L2,L3,Address,Poctcode,Lat,Lon,Capacity,Population,Status  
COC_nome,,Roma_PalaLottomatica,IT,Lazio,Roma,Roma,Piazzale Pier Luigi Nervi 1,00144,41.825213,12.465835,,,Open
```

HOUSING UNITS

A shelter must be considered as a container inside which there the “Housing Units” are defined. The SW manages the shelters and the Housing Units as different type of objects: a shelter could be a temporary facility or a hotel; the housing units are instead tents or rooms in a hotel.

The housing unit registration is implemented inside the shelter module (CR) and in particular inside the sheet “Housing Units”. When a new Housing Unit is registered, some actions and checks are automatically run to update the total capacity of the associated shelter (available in the tab “Basic Details”) and the global table where all the shelters are registered.

Each Housing Unit displays some properties that describe its characteristics, such as, for example: the hosting capacity, the services available, such as toilets or showers for disabled and so on.

Evacuees registration inside a specific Housing Unit

EVASS considers not only the possibility of registering evacuees into a shelter, but it also offers the option to register them into a specific housing unit belonging to a shelter, in order to provide a better and ordered management.

The action to register a new evacuee inside the shelter and then to the related housing unit can be submitted from the Evacuees module (“Shelter Registration” tab) or from the Shelter module (“People Registration” tab).

The registration tab offers the possibility to register an evacuee inside a shelter or at the same time the evacuee can be registered into the shelter and housing unit. The two actions are independent and can be run also at separate times; the evacuee can be registered into a shelter, and then at a later moment, when the complete picture of the evacuees is clear, they can be assigned to a specific housing unit.

The evacuee’s registration (to both shelter or housing unit), automatically updates the current hosted population. The system can check the size of the current population with respect to what was declared as the general capacity of the shelter or housing unit; if the limit is exceeded, a no blocking popup notifies the situation. This means that, if needed, the owner can continue to register new evacuees even if the limit has been reached (because it might be possible to redirect people to other shelters).

EVACUEES

EVASS includes functions to manage evacuees during an emergency or after the emergency. The dedicated module is Evacuees (evr) but the management can also be done through the module Shelters (as explained in the previous section “[Shelter](#)”).

Each single Evacuee is identified by many different properties, which have been aggregated into sheets, according to the type of information. The sheets currently active in EVASS are:

Person	Include generic person’s information such as identity record, organizations that owned the person, social status, and so on.
Addresses	Includes information useful to find the person such as home address or address of the company where he works, and so on
Contact Data	Includes information useful to contact the person (such as, phone number, email and so on)
Images	Photograph of the person to support an easy identification
Groups	Includes information about the groups to which the person belongs (family, friends, tourist group, and teamwork)
Medical Information*	Includes the state of health of the person
Socio – Economic Background	Includes generic information about the social state of a person (required for appropriate assistance).
Shelter Registration	To register the evacuee into a shelter.

**ONLY the users with “Medical” or “Administrator” role have access to this set of data.*

EVACUEES registration

To assist the population affected by an emergency, each person must be registered into the SW and this can be done by using two methods: manually or through a CSV file.

Manual registration

The Administrator or each user authorized to use the SW can manually insert the evacuees using the module “Evacuees”. To start the insertion, select “New” in the menu or click the “Create a person” button.

There is a basic sheet “Person” where the user can enter the basic data of the person (mandatory phase); other data (“additional sheets”) can be provided later

NOTE:

The field “Organization” means the organization that owns the evacuee; this field is not mandatory but it must be used to include the evacuee in the specific domain (for details see “Roles and Domains”). Assigning a person to a specific organization gives access to view the data only to those users belonging to the same organization.

If NOT specified, the only users who can access the data of the unassigned evacuee are the

SW Administrators.

Registration through a CSV file

The CSV import is the method to automatically register more than one evacuee with only one operation. This action is available in the module Evacuees where the button Import is available. The CSV file to import must follow a template that can be downloaded from the SW (a link is available in the “Download Template”). The fields to fill in the template are:

Organization	Optional	Organization or branch committed to provide assistance. *
First Name	Mandatory	Name
Middle Name	Optional	Second Name
Last Name	Mandatory	Family Name
Date of Birth	Mandatory	Date of birth: Format: mm/dd/yyyy m/d/yyyy mm,dd,yyyy mm-dd-yyyy
Place of Birth	Optional	Place of birth
Gender	Mandatory	Sex. Male, Female. Format: M, F,
Fiscal Code	Optional	Tax Code
Home Address	Optional	Address of Residence
Home Postcode	Optional	Postal Code
Home L1**	Optional	Region of Residence
Home L2**	Optional	District of Residence
Home L3**	Optional	City of Residence
Home L4**	Optional	Community / area of Residence
Home Phone	Optional	Fixed phone. The format is the international: eg. +39 123456
Mobile Phone	Optional	Mobile phone. The format is the international: eg. +39 123456

* The organization name must be the same as an organization already defined and registered in the SW (case insensitive). If the name does not exist, a new organization will be created.

** The location fields must be completely compiled; the SW discards the template automatically when only one field is provided (example L3). This is an example of a CSV file:

```
Organisation,First Name,Middle Name,Last Name,Date of Birth,Place of Birth,Gender,Fiscal Code,Home Address,Home Postcode,Home L1,Home L2,Home L3,Home L4,Home Phone,Mobile Phone
Nome_org,Mario,Marco,Rossi,01/21/99,Monopoli,M,MRCRSS99A01F376N,Via Roma 33,70043,Fuoglia,Bari,Monopoli,,+39 4444470,+39 222248
```


MAPPING / GEO Localization

Sahana Eden includes mapping functionalities in order to make easy, fast and integrated the management of the emergency. Much of the data available in the database can be represented on the map.

The map's functionalities are mainly collected inside the "gis" module that, as any other module, can be activated or deactivated by editing the configuration file: `private/template/EVASS/config.py` (in EVASS it is activated by default).

The system works with a basic map called "base layer". On top of this, the users can define new layers of information (multiple overlapping layers). Each layer internally keeps many different types of data, which is that available in the database (shelters, staff, volunteers, and so on).

EVASS currently selected as "base layer" the following maps:

- Google⁵ Map Terrain*
- Google Map Hybrid*
- Google Maps*
- OpenStreetMap Humanitarian (Default)
- OpenStreetMap MapQuest
- OpenStreetMap Standard

This set of maps can be interchanged with no limitations, but it is always recommended to select the correct map according to the specific needs, because each one contains different information that can support the Organization's team.

Many other Information layers can be included on top of the "base layer".

Currently EVASS has defined the following layers:

PointOfInterest	Show the Point of Interests defined by the users
Incidents	Show the active incidents defined inside the events
Staff	Show the Staff members assigned to the shelters
Volunteers	Show the Volunteers assigned to the shelter
Shelters	Show the Evacuees shelters in state "Open"
Offices	Show the Organization's office. "Unused" offices are not shown.
Facilities	Show the Organization's Facilities. "Unused" facilities are not shown.

EVASS has defined also layers concerning Online Services:

5

Google maps are available only when a direct connection to Internet is available.

Earthquakes	Show the most serious Earthquakes that occurred recently in the world. It uses as reference the server: http://earthquake.usgs.gov/
Population Density	Show a map of the population density in the world (low resolution map). It uses as reference the server: http://sedac.ciesin.columbia.edu/
Current Weather	Show the real-time weather. It uses as reference the server: http://openweathermap.org/
Weather Stations	Show the weather stations involved in the real-time weather forecast. It uses as reference the server: http://openweathermap.org/
Cloud Forecast	Show the Cloud forecast and its density
Precipitation Forecast	Show the Precipitation Forecast

The current settings allow access to the maps to all the users previously authorized to use the SW application.

The data available on the map is associated to the role and realm of the users using the SW application.

The module “Map” includes also a search service (on top of the map area), which is useful when there is the need to focus on a specific area of interest (country, district or city). Invoking an external free service named “Geonames” satisfies the search requests. The Geonames service is not available when Sahana Eden starts for the first time; it requires a configuration step to be run. To correctly use this service:

1. Register to the web site <http://www.geonames.org/>
2. Look for the section “manage account” and activate the additional services. Account description must contain the string “the account *Account_Username* has been enabled for the free web services”.
3. Update the file Private/Template/EVASS/Config.py including the username specified during the registration to Geonames.

```
settings.gis.geonames_username = "Account_Username"
```

4. Restart the Server

How to configure the Map

The Map configuration is deferred only to the role of “Administrator” or “Map Admin”; they can change the registered information.

In accordance with the Italian Organization, EVASS has activated only some geographical information and few levels of the available maps.

In the section “Location” are defined all the Italian local governments that follow a hierarchical organization based on layers:

- [Country – Layer 0 \(L0\)](#)

- Region – Layer 1 (L1)
- District – Layer 2 (L2)
- Town – Layer 3 (L3)

L0, L1 and so on are used to represent the hierarchical layers.

At the first startup of the application, EVASS automatically initializes the hierarchy including all the administrative divisions active in Italy (region, district/province, city).

It's always possible extend and define new locations, but it is important to maintain the correct hierarchical layer of membership.

In the section “Configuration” it is possible to define a personalized configuration of the map (zoom, central position, projections). The “Default” configuration includes all the needed setup for a complete and correct operation. This setup is populated when the application starts for the first time.

“Admin” includes the following sections:

1. Hierarchy: Configurable section including the hierarchical organization of the Italian local government. Populated when the Application starts for the first time.
2. Layers: Use this section to Define / Delete new layers of information to display on the map. Two types of information can be configured:
 - Information available in the local database (select Features and then Internal Resource)
 - Information coming from external sources or server. Sahana makes available the possibility to interface servers implementing the following technologies: ArcGIS, GeoJSON, GeoRSS, GPX, KML, MGRS, OSM, OWM, Shapefile, TMS, WFS, WMS, XYZ tiles
3. Markers: This section includes the list of icons to display the information on the map. Populated when the Application starts for the first time.
4. Pol Types: This section includes all the types of Point of Interest. Populated when the Application starts for the first time.
5. Projections: This section includes the mathematical algorithms applied to transform the geographical points on the map (from geographical coordinates to Cartesian coordinates). Populated when the Application starts for the first time.
6. Symbology: This section includes a set of icons available for some particular deployments. Populated when the Application starts for the first time.

When the Application starts for the first time, EVASS populates all the layers representing the hierarchy applicable to the Italy security model that is reflected also in the operational model.

CONFIGURATION

As anticipated in the previous chapters, Sahana Eden is software that can be heavily customized and configured. For this reason it covers many different needs and can be adapted to the requirements of the end user. This section summarizes the complete configurations steps. The configuration is done via two text files, located in the working directory:

- [modules/000_config.py](#) for generic configurations
- [private/templates/EVASS/config.py](#): it has been created for Italy-related configurations (in general, if there's no reason to do otherwise, this should be the preferred one). This configuration file is recommended for those organizations wanting to use EVASS.

Configuring [modules/000_config.py](#)

The file [modules/000_config.py](#) is the main file for configuring Sahana Eden for generic purposes. It drives the overall Sahana and its parameters, if modified, can change the standard behavior of some functions.

The file [modules/000_config.py](#) is not available by default (when the code is downloaded) but is generated after starting Sahana for the first time. The initial setting is not valid for a correct functioning of Sahana EVASS; therefore it must be configured properly. To use the EVASS template, the following setting is recommended:

1. Start Sahana Eden for the first time and locate the file: [modules/000_config.py](#)
2. Remove the following line: `FINISHED_EDITING_CONFIG_FILE = False`
3. Provide the template to use: by default the file specifies as template the "Default", but for Italy the template to activate is "EVASS". The row to change is:
from settings.base.template = "default" to settings.base.template = "EVASS"
4. The file includes the public IP of the Sahana server. It is provided to all the users who will connect to Sahana from outside the Intranet `settings.base.public_url = http://Sahana_public_IP_address:8000 Example:"http://10.121.132.4:8000"`
5. Configure email interfaces to send email notification. To activate the changes, restart the servers or reload the browser. Below there are two different mail systems configurations (GMAIL and Hotmail), but other mail systems are allowed:

Gmail Configuration:

#*****Email settings*****	
settings.mail.mailsubject = "Sahana Eden for Italy"	
settings.mail.approver = " example@example.com "	i.e " admin@gmail.com "
#settings.mail.limit = 1000	
settings.mail.server = "mail_server:port"	i.e. "smtp.gmail.com:587"
settings.mail.tls = True	
settings.mail.login = " example@gmail.com:password "	i.e. " admin@gmail.com:pwd "
settings.mail.sender = "'Sahana' Administrator"	

Hotmail/Live Configuration:

#*****Email settings *****	
settings.mail.mailsubject = "Sahana Eden for Italy"	
settings.mail.approver = " example@example.com "	i.e. " admin@live.com "
#settings.mail.limit = 1000	
settings.mail.server = "mail_server:port"	i.e. "smtp.live.com:587"
settings.mail.tls = True	
settings.mail.login = " example@live.com:livepassword "	i.e. " admin@live.com:livepwd "
settings.mail.sender = " example@example.com "	i.e. " admin@live.com "

Configuring private/templates/EVASS/config.py

This file must be considered to be the main configuration file when the template EVASS is selected. All the settings and changes applied to this file are valid only internally to EVASS**.

There are many parameters that can be configured and those listed below are only a subset of them; the complete list and their description is available in the file [/private/template/default/config.py](#)

The file available in the SW has been already configured to respond to the needs of Italian organizations. In general there is no need to change it unless there are some specific needs. The complete description of each parameter is available in the file

/private/template/default/config.py. After changing the settings, restart the server to activate the changes.

** The configurations defined in the private/templates/EVASS/config.py are considered only if the file modules/000_config.py has been properly configured. An example of the file already configured in EVASS is available in [Appendix B](#).

Map settings

GeoNames username

```
settings.gis.geonames_username = "*****"6
```

As explained before, in the module of mapping, there is a feature available (geonames) that submits requests towards an external service but, in order to use it, the user must follow a registration process on the web site: <http://www.geonames.org/> and activate all the requested advanced options in the profile created before.

The username that is created will be used to properly set the parameters in the GeoNames section.

Restrict the Location Selector to just certain countries

```
settings.gis.countries = ["IT"]
```

Display the Map Legend as a floating DIV

```
settings.gis.legend = "float"
```

```
settings.pr.show_emergency_contacts = False
```

Link evacuees to Organisations

```
settings.evr.link_to_organisation= True
```

6

The usage of geonames depends on an external service. To use it, you must be registered on the web site: <http://www.geonames.org/> and have to set up the additional options in your profile. The user account must be able to use the web services.

STARTING EDEN

Sahana Eden is now installed and properly configured. To start the application and make it visible to the end users:

On Windows systems:

Move to the folder private/templates/EVASS/Web2py and start the application “web2py.py” (using python interpreter)

On Linux systems:

Using the Terminal, move to the folder private/templates/EVASS/Web2py and start the application “web2py.py” by using the command:

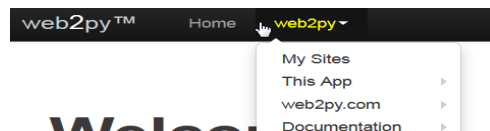
`python web2py.py`

Select the type of service to start:

- Local: private and not reachable by other users
- Public: visible and available to other users who know the ip address



Select “My Sites”



and then EDEN.

Login

During the generation and population of the database tables, Sahana Eden automatically creates two default users to allow the initial login and to use the software:

Username	Password
admin@example.com	testing
normaluser@example.com	testing

The user admin@example.com is assigned with the role of “administrator” and has access to all the resources and functions of the software.

At the first Login, the following actions are strongly suggested:

1. Access Sahana Eden using the default administrator user “admin@example.com”
2. Create one or more users and assign the role of “Administrator”
3. Log out and Log in to the software using the new created accounts in order to validate them.
4. Disable the default users admin@example.com and normaluser@example.com (from Administration, click Users and then Disable)

The main page of the EVASS template is organized into three parts and appears as follows:

EVASS Map Organizations Messaging Administration admin@example.com Language

Welcome to EVASS!

Events

Shelters Evacuees

Person Registry Staff Volunteers

News

@crocerossa. Presidente #crocerossa @Francescorocca: #CRI è neutrale. Al razzismo 2.0 consiglio di studiare di più' <http://bit.ly/1wDLYWw> @LBastianetto
17 Set 2014
Presidente #crocerossa @Francescorocca: #CRI è neutrale. Al razzismo 2.0 consiglio di studiare di più' <http://bit.ly/1wDLYWw> ...

Top layer includes the following buttons:

EVASS	Link to the homepage.
Map	Link to the map module (gis)
Organizations	Link to the organizations module (org)
Messaging	Link to the notification module (msg)
Administration	Link to the module to administer the application
User Button	Link to the functions available to the final user of the system
Language	Select the language. Available languages are English and Italian.
Help	Useful link and support

Middle layer includes the following buttons:

Events	Link to the module to manage the events (event)
Shelters	Link to the module to manage the homeless shelters (cr)
Evacuees	Link to the module to manage the evacuees (evr)
Person Registry	Link to the human resource registry (pr)
Staff	Link to the module to administer the staff (hrm/staff)
Volunteers	Link to the module to administer the volunteers (vol)

Bottom layer in which are located RSS information coming from the Italian Department of Civil Protection and Tweets from the official Twitter of the Italian RED Cross.

Users Registration

Each user registered in Sahana Eden is already authenticated and can access a limited set of data available in the database.

The type of data strictly depends on the “Role” and “Realm” assigned by the administrator.

There are three methods to register new application users:

1. System Administrator manually registers the new application users.
2. Automatic import of a CSV file containing the data of the new users to be registered.
3. Application User’s self-registered inside the SW.

EVASS provides only Methods 1 and 2. Method 3 can be enabled inside `Private/Template/EVASS/config.py` (if requested)

System Administrator registers a new user

To start using the application and define some resources, the Administrator should log in to the system and create a default Organization that will act as a container for all the resources / assets that need to be registered into the system as permanent resources involved in any emergencies / events that will be managed by Sahana Eden. To create a default organization:

- a. From the Main page, Select Organization
- b. Create a new Organization

It is now possible to continue the registration process.

To register a new user, the Administrator should log in to the system and run the following steps:

- c. From Administration, select Users
- d. From User Management, select Create User
- e. Fill in the data of the new user
- f. Provide the “Role” and “Realm” to be associated to the new user.

The new user is now authenticated and authorized to use the SW and to manage the data according to his profile.

Registration through a file

When there are many users to be registered simultaneously (i.e. when the tool is initially configured inside an organization), it is possible to import the new users through a CSV file.

To start this procedure, the Administrator should log in to the system and run the following steps:

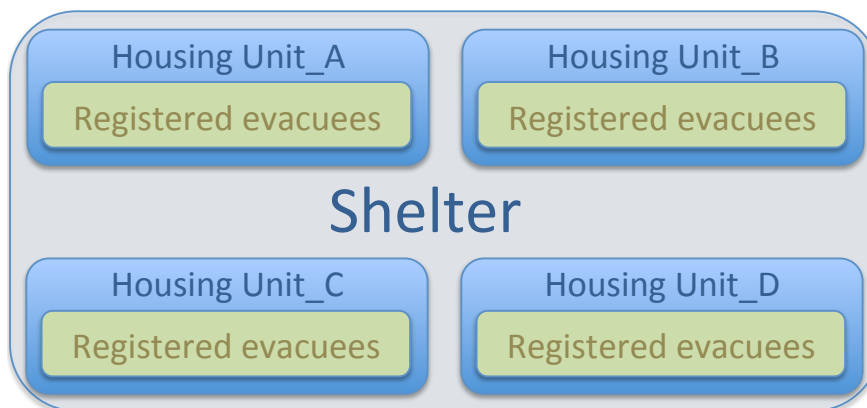
- a. From Administration, select Users
- b. From User Management, select Import Users

The CSV file must be generated before following the suggested template.

The new users included in the CSV file, are now authenticated and authorized to use the SW and to manage the data according to their profiles.

NOTA

It is very important to check the correct Organization and Domain / Realm assigned to each new user.



REPORTING

The platform Sahana Eden includes some reporting functionality.

Currently it is possible to extract some information from the modules shelters, evacuees, staff and volunteers. The information could be global if we are extracting full data in the table or partial when the data is filtered and combined using the available filters.

The available reporting system can create a two-dimensional table where the user must specify which kind of data must be represented on the X-axis and Y-axis.

The reports can also be represented as pie charts or histograms.

USING EVASS - tips

Sahana Eden is a Web-based SW available through the common browser.

Start the SW as described in the chapter and then point the browser to:

1. <http://127.0.0.1:8000/eden> (for a local instance)
2. http://server_ip_address:8000/eden (for a common and public instance)

All the remote clients can reach the public instance of Sahana by pointing the browser to:

http://server_ip_address:8000/eden

Using multiple instances of EVASS

Sahana Eden can run multiple independent instances of the same installation. The different instances can be joined later: both instances and associated database will be merged and run as only one instance.

The merge is an Administrative task and can be submitted using the advanced feature “Synchronization”, available in the menu Administration.

All the information and guides to use Sahana in this configuration are available on the official wiki: <http://eden.sahanafoundation.org/wiki/UserGuidelines/Admin/Synchronization>

NOTE:

EVASS should benefit from the original synchronization feature available in Sahana EDEN. The EVASS template did not run any additional tests using this configuration with distributed instances and databases.

Component's Dependencies

EVASS and its components (new and existing) are all part of Sahana Eden, so their dependencies are exactly the same as the main software:

<http://eden.sahanafoundation.org/wiki/InstallationGuidelines/Windows/Developer/Manual>

Components have dependencies from many external libraries and tools. Some of them are listed below.

Mandatory:

- Python 2.7 is currently recommended. (2.6 works for all core functionality, but doesn't allow you to use the S3Report pivot table functions). Note: 32-bit Python should be used even on 64-bit Windows
- Web2py web framework
- lxml for XML export/import
- Shapely for GIS Features
- setup tools for installing dateutil (& tweepy)
- dateutil

Optional:

- xlrd for XLS output
- xlwt for XLS output
- ReportLab for PDF output
- PIL Python Image Library for PDFs
- GDAL for Shapefile support
- PyRTF for Survey to eXport as RTF:
- Matplotlib for charts (used in the Survey application but not for the S3Report pivottable reports)
- Numpy required by matplotlib & also Vulnerability module
- IPython for a nicer interactive shell
- Python eXtensions for Windows
- tweepy python library for Twitter Send/Receive
- TwitterSearch python library to search Twitter
- Pyserial to use a Modem for SMS
- RPy2 for Climate module
- SciPy for Climate module

- NetCDF for Climate module
- MySQLDB for accessing MySQL databases
- Psycopg for accessing PostgreSQL databases
- PostgreSQL-9.2.4

List of components

In order to deliver the above set of scenarios, the project has created new components but also modified existing ones (keeping backward compatibility). By configuring the EVASS template properly, only some functionality are available through the home page and are those identified by Italian organizations as the most relevant to manage the evacuees. Enabling the other components (available in Sahana) is always possible but requires some additional tests (it's enough to customize the configuration file). Here below is the list of components in EVASS:

- Evacuees Registry (evr): new module used to store data of the people affected by the emergency.
- Evacuees ASSistance (EVASS): a new template specifically created for Italy where all the specific bits and customizations are placed to best suit the Italian needs.
- Camp Registry (cr): a module that contains information about the camps, shelters and in general the facilities where people can to be located in response of an emergency. It was part of Sahana Eden but has been modified to accommodate EVASS requirements.
- Event (event) and the related Incidents: to keep track of the emergency details.
- Components to register and manage volunteers or groups of volunteers, members of staff, and so on.
- Organizations (ORG): to define the organizations hierarchy managing the emergency and authorized to work on it and properly manage the associated data (in respect of the law and data privacy).
- Notifications (msg): to manage the communication among involved employees or systems. Notification can be email to person or group, SMS, or using social tools (if configured).
- Geolocalization (Gis): to include the support of maps.

Component High-Level design

The new components are designed with compatibility and interoperability versus the already existing Sahana platform. Each component follows the Model View Controller pattern.

a.1 Evacuees Registry follows this principle by inheriting the view from the default Sahana Eden template, and implementing a completely new:

- Model for data storage
- Controller to interact with the Model through the UI

It basically extends the pre-existing Person Registry module, by adding a number of additional pieces of information that would help the Shelter staff to allocate the person and his/her family or group to the best available accommodation.

The application has implemented a mechanism to unique identify evacuees and remove any possible duplicated records: it is based on the “Fiscal Code”⁷, a mechanism available in Italy that assigns a personal identification number to every citizen and is unique. The field “Fiscal Code” is not mandatory because it is possible that during an emergency people don’t have their documents with them (at least in the first stage of the registration). There are two possible cases:

- a. Evacuees bring document and provide Fiscal Code – automatically check is run to grant the uniqueness. This allows a control over erroneous or malicious record duplication in the system.
- b. Evacuees do not have document and Fiscal Code is not specified - the duplication check must be run manually.

Fiscal code consistency is checked only against the following fields:

Name, Surname, Date of birth, Gender.

The implemented check is only partial and not exhaustive because it is based only on 4 fields that nevertheless are relevant, but are only representing part of the entire Fiscal Code. It has been provided for convenience.

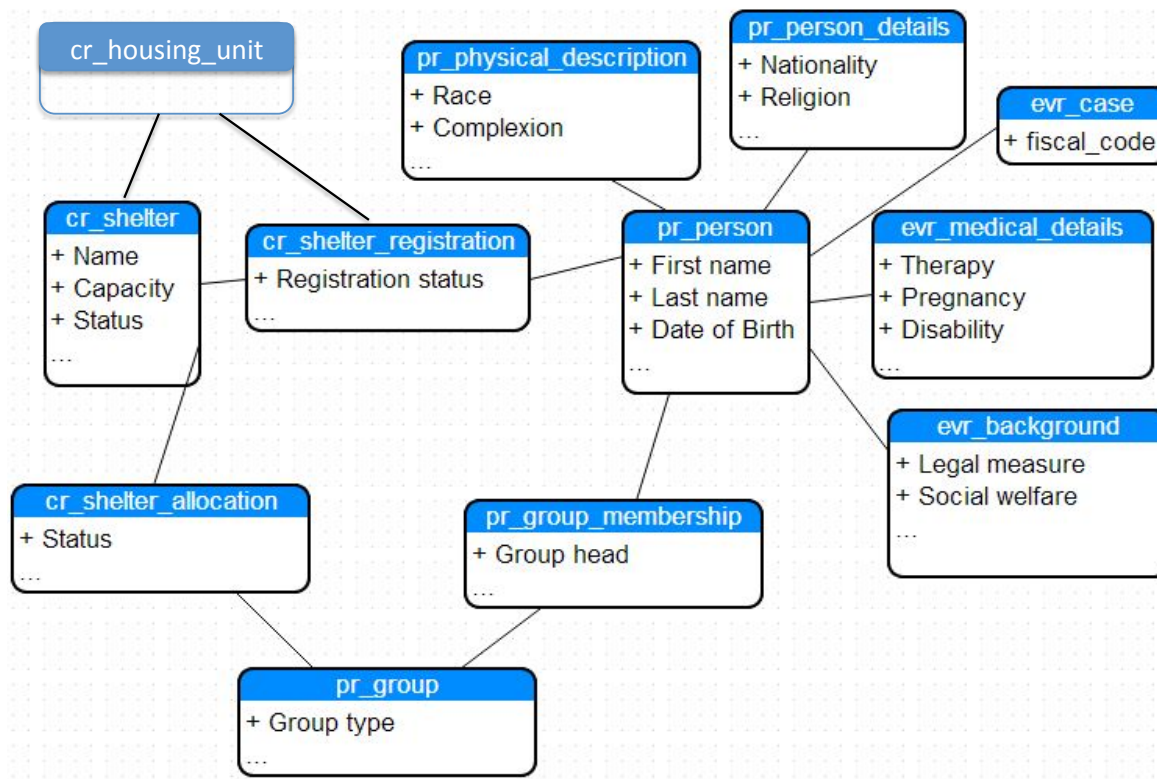
- b.1** Camp Registry has been heavily modified to permit a tight interaction with Evacuees Registry by creating a new logic that links those modules. Shelter population number is now calculated from the currently registered people and from the general registered capacity; the tool offers also the possibility to allocate or reserve seats for future registration. Using all that information, logic has been implemented to calculate the available capacity for a specific shelter. The shelter does not physically contain the persons but is a container for the Housing Units, where the people receive accommodation
- c.1** Shelters Housing Units is a location inside the shelters where the persons live during their stay. Through the housing Units, the personnel can manage the evacuees by tracking all the related information, keeping track of their movements and registrations. The shelter is like a container composed of many housing Units; the characteristics of a shelter must be intended as the sum of the characteristics of all the housing units. The “real capacity” of a shelter, that is the effective number of persons living there, is considered to be the sum of the capacity of all the housing units defined in the shelter. Each housing unit keeps track of the general capacity and the effective number of registered persons and dynamically calculates the current availability (for the night or for day and night).

7

Fiscal Code identifies every tax payer. The Information System automatically assigns the Fiscal Code to all citizens. For a complete explanation : [Agenzia delle Entrate](#)

DATABASE CHANGES / EXTENSIONS

The basic database representation has been extended following this ER:



New tables have been created:

- evr_case
- evr_medical_details
- evr_background
- cr_shelter_registration
- cr_shelter_allocation
- cr_housing_unit

Other tables have been integrated with further information needed for Italy:

- pr_person_details
- pr_physical_description
- cr_shelter

Considering that:

- pr_person is the main table and contains generic personal data
- pr_physical_description includes information to easily identify the person
- pr_person_details extends the generic information found in pr_person
- evr_case contains the Fiscal Code of the evacuee and identifies him

- `evr_medical_details` contains medical information that can be used by application users to correctly select the right accommodation.
- `evr_background` contains legal and generic information about the evacuee
- `pr_group` permits the specification of families or other type of groups of people; it is double-linked with `pr_person` through `pr_group_membership`
- `cr_shelter` represents the facility that hosts people and contains all the information about the shelter details and status. It's linked with people by the `cr_shelter_registration` table and with groups of people by `cr_shelter_allocation`.

APPENDIX A - Deliverables

When the EVASS template is enabled, this is a list of the modules mandatory for a correct use of EVASS. The table includes also a list of the functions currently available; some of them have been activated because applicable to the Italian scenarios, some others have been disabled.

They can be enabled in the future but additional tests are needed.

Mandatory Modules	Functions enabled	Functions disabled
"Home" "Administration" "Ticket Viewer" "Synchronization" "Translation Functionality"	"Map" "Person Registry" "Organizations" "Staff" "Volunteers" "Documents" "Messaging" "Shelters" "Evacuees" "Events"	"Guided Tour Functionality" "Support" "Content Management" "Supply Chain Management" "Warehouses" "Procurement" "Assets" "Vehicles" "Requests" "Projects" "Surveys" "Hospitals" "Incidents" "Disaster Victim Identification" "Disaster Victim Registry" "Missing Person Registry" "Transport" "Statistics" "Scenarios" "Vulnerability" "Fire Stations" "Flood Warnings" "Members" "Patient Tracking" "Security" "CAP Alert" "Climate" "Delphi Decision Maker" "Building Assessments" "Assessments" "Impacts" "Optical Character Recognition"

APPENDIX B – Example of private/templates/EVASS/config.py

```

# -*- coding: utf-8 -*-

try:
    # Python 2.7
    from collections import OrderedDict
except:
    # Python 2.6
    from gluon.contrib.simplejson.ordered_dict import OrderedDict

from gluon import current
from gluon.storage import Storage
from gluon.validators import IS_NOT_EMPTY, IS_EMPTY_OR, IS_IN_SET
from s3 import s3_date, S3Represent

T = current.T

settings = current.deployment_settings

"""
Settings for the EVASS template:
http://eden.sahanafoundation.org/wiki/Deployments/Italy/EVASS
"""

# -----
# Pre-Populate
settings.base.prepopulate = ["EVASS", "demo/users"]
settings.base.system_name = T("EVASS - Sahana Eden for Italy")
settings.base.system_name_short = T("Sahana Eden for Italy")
# Theme (folder to use for views/layout.html)
settings.base.theme = "EVASS"
settings.ui.formstyle = "foundation"
settings.ui.filter_formstyle = "foundation_inline"
settings.ui.hierarchy_theme = "default"

# -----
# Email settings
settings.mail.default_email_subject = True
settings.mail.auth_user_in_email_subject = True

```

```
# -----
# Authentication settings
settings.auth.registration_roles = {"organisation_id": ["CR_EVR_READER_V",
],
}
settings.auth.registration_requests_mobile_phone = True
settings.auth.registration_mobile_phone_mandatory = True
settings.auth.registration_requests_organisation = True
# Uncomment this to have the Organisation selection during registration be mandatory
#settings.auth.registration_organisation_required = True
settings.auth.always_notify_approver = False
settings.security.self_registration = False
# Security Policy
# http://eden.sahanafoundation.org/wiki/S3AAA#System-widePolicy
settings.security.policy = 7

# -----
# L10n settings
settings.L10n.languages = OrderedDict([
("en", "English"),
("it", "Italiano"),
])
settings.L10n.default_language = "en"
settings.L10n.utc_offset = "UTC +0100"
settings.L10n.date_format = T("%d/%m/%Y")
settings.L10n.decimal_separator = ","
settings.L10n.thousands_separator = "."
settings.L10n.default_country_code = 39
settings.L10n.mandatory_lastname = True
settings.L10n.translate_gis_location = True
# Finance settings
settings.fin.currency_default = "EUR"
settings.fin.currencies = {
"EUR": T("Euros"),
```


“GBP”: T(“Great British Pounds”),

“USD”: T(“United States Dollars”),

}

GIS (Map) settings

GeoNames username

settings.gis.geonames_username = “geoname_username”

settings.gis.countries = [“IT”]

settings.gis.legend = “float”

settings.gis.nav_controls = False

Shelters

settings.cr.shelter_population_dynamic = True

settings.cr.shelter_housing_unit_management = True

Events

settings.event.types_hierarchical = True

Evacuees

settings.evr.physical_description = False

settings.pr.show_emergency_contacts = False

settings.evr.link_to_organisation= True

Organisations

settings.org.branches = True

settings.org.branches_tree_view = True

settings.org.facility_types_hierarchical = True

Human Resource Management

settings.hrm.email_required = False

settings.hrm.org_required = False

settings.hrm.deletable = True

settings.hrm.multiple_job_titles = True

```

settings.hrm.staff_experience = False
settings.hrm.vol_active = True
settings.hrm.vol_experience = False
settings.hrm.show_organisation = True
settings.hrm.use_awards = False
settings.hrm.use_certificates = False
settings.hrm.use_skills = True
settings.hrm.use_trainings = False
#*****Frontpage settings*****
# RSS feeds
settings.frontpage.rss = [
{"title": "RSS News - Dipartimento della Protezione Civile ",
"url": "http://www.protezionecivile.gov.it/jcms/do/jprss/Rss/Feed/show.action?id=12170&lang=it#"
},
{"title": "RSS Vigilanza Meteo - Dipartimento della Protezione Civile ",
"url": "http://www.protezionecivile.gov.it/jcms/do/jprss/Rss/Feed/show.action?id=23573&lang=it#"
},
{"title": "RSS Previsioni Meteo - Dipartimento della Protezione Civile ",
"url": "http://www.protezionecivile.gov.it/jcms/do/jprss/Rss/Feed/show.action?id=23575&lang=it#"
},
{"title": "RSS Comunicati Stampa - Dipartimento della Protezione Civile ",
"url": "http://www.protezionecivile.gov.it/jcms/do/jprss/Rss/Feed/show.action?id=23577&lang=it#"
},
},
{"title": "Twitter - Croce Rossa Italia",
# @crocerossa
#"url": "https://search.twitter.com/search.rss?q=from%3Acrocerossa" # API v1 deprecated, so doesn't work,
need to use 3rd-party service, like:
"url": "http://www.rssitfor.me/getrss?name=@crocerossa"
# Hashtag
#url: "http://search.twitter.com/search.atom?q=%23eqnz" # API v1 deprecated, so doesn't work, need to use
3rd-party service, like:
#"url": "http://api2.socialmention.com/search?q=protezionecivile&t=all&f=rss"
},
# {"title": "Twitter - Dipartimento della Protezione Civile",
# # @protezionecivile

```

```

# "url": "http://www.rssitfor.me/getrss?name=@protezionecivile"

# # Hashtag

# #url: "http://search.twitter.com/search.atom?q=%23eqnz" # API v1 deprecated, so doesn't work, need to use
3rd-party service, like:

# "url": "http://api2.socialmention.com/search?q=protezionecivile&t=all&f=rss"

# }

]

# -----
def customise_pr_person_resource(r, tablename):

s3db = current.s3db

table = r.resource.table

# Disallow "unknown" gender and defaults to "male"

evr_gender_opts = dict((k, v) for k, v in s3db.pr_gender_opts.items())

if k in (2, 3))

gender = table.gender

gender.requires = IS_IN_SET(evr_gender_opts, zero=None)

gender.default = 3

if r.controller == "evr":

# Hide evacuees emergency contacts

settings.pr.show_emergency_contacts = False

# Last name and date of birth mandatory in EVR module

table.last_name.requires = IS_NOT_EMPTY(error_message = T("Please enter a last name"))

dob_requires = s3_date("dob",

future = 0,

past = 1320,

empty = False).requires

dob_requires.error_message = T("Please enter a date of birth")

table.date_of_birth.requires = dob_requires

# Enable Location_id

from gluon import DIV

from s3.s3widgets import S3LocationSelectorWidget2

levels = ("L1", "L2", "L3",)

location_id = table.location_id

```

```
location_id.readable = location_id.writable = True

location_id.label = T("Place of Birth")

location_id.widget = S3LocationSelectorWidget2(levels=levels,
lines=True,

)
location_id.represent = s3db.gis_LocationRepresent(sep=" | ")

# Enable place of birth
place_of_birth = s3db.pr_person_details.place_of_birth
place_of_birth.label = "Specify a Different Place of Birth"
place_of_birth.comment = DIV(_class="tooltip",
_title="%s|%" % (T("Different Place of Birth"),
T("Specify a different place of birth (foreign country, village, hamlet)")))
place_of_birth.readable = place_of_birth.writable = True

# Disable religion selection
s3db.pr_person_details.religion.readable = False
s3db.pr_person_details.religion.writable = False

# Disable unneeded physical details
phtable = s3db.pr_physical_description
hide_fields = [
"race",
"complexion",
"height",
"weight",
"hair_length",
"hair_style",
"hair_baldness",
"hair_comment",
"facial_hair_type",
"facial_hair_length",
"facial_hair_color",
"facial_hair_comment",
"body_hair",
```

```
“skin_marks”,
“medical_conditions”

]
for fname in hide_fields:
field = pdtable[fname]
field.readable = field.writable = False
# This set is suitable for Italy
ethnicity_opts = (“Italian”,
“Chinese”,
“Albanese”,
“Philippine”,
“Pakistani”,
“English”,
“African”,
“Other”,
“Unknown”,
)
ethnicity_opts = dict((v, T(v)) for v in ethnicity_opts)
ethnicity = pdtable.ethnicity
ethnicity.requires = IS_EMPTY_OR(IS_IN_SET(ethnicity_opts,
sort=True))
ethnicity.represent = S3Represent(options=ethnicity_opts,
translate=True)
settings.customise_pr_person_resource = customise_pr_person_resource
def customise_cr_shelter_resource(r, tablename):
s3db = current.s3db
s3db.cr_shelter.capacity_day.writable = s3db.cr_shelter.capacity_night.writable = False
s3db.cr_shelter.cr_shelter_environment_id.readable = s3db.cr_shelter.cr_shelter_environment_id.writable =
True
settings.customise_cr_shelter_resource = customise_cr_shelter_resource
def customise_pr_group_resource(r, tablename):
messages = current.messages
field = r.table.group_type
```

```

pr_group_types = {1 : T("Family"),
2 : T("Tourist Group"),
3 : T("Relief Team"),
4 : T("other"),
5 : T("Mailing Lists"),
6 : T("Society"),
}
field.represent = lambda opt: pr_group_types.get(opt, messages.UNKNOWN_OPT)
field.requires = IS_IN_SET(pr_group_types, zero=None)
settings.customise_pr_group_resource = customise_pr_group_resource

# -----
def customise_event_event_resource(r, tablename):

table = r.table

table.exercise.default = True

table.organisation_id.readable = table.organisation_id.writable = True
settings.customise_event_event_resource = customise_event_event_resource
def customise_event_incident_resource(r, tablename):

table = r.table

table.exercise.default = True

table.event_id.readable = table.event_id.writable = True
settings.customise_event_incident_resource = customise_event_incident_resource

# -----
def customise_project_location_resource(r, tablename):

field = current.s3db.project_location.status_id

field.readable = field.writable = True

settings.customise_project_location_resource = customise_project_location_resource

# -----
# Comment/uncomment modules here to disable/enable them
# @ToDo: Have the system automatically enable migrate if a module is enabled
# Modules menu is defined in modules/eden/menu.py

settings.modules = OrderedDict([

# Core modules which shouldn't be disabled

("default", Storage(
name_nice = T("Home"),

```

```
restricted = False, # Use ACLs to control access to this module

access = None, # All Users (inc Anonymous) can see this module in the default menu & access the controller
module_type = None # This item is not shown in the menu

)),
("admin", Storage(
name_nice = T("Administration"),
#description = "Site Administration",
restricted = True,

access = "|1|", # Only Administrators can see this module in the default menu & access the controller
module_type = None # This item is handled separately for the menu

)),
("appadmin", Storage(
name_nice = T("Administration"),
#description = "Site Administration",
restricted = True,

module_type = None # No Menu

)),
("errors", Storage(
name_nice = T("Ticket Viewer"),
#description = "Needed for Breadcrumbs",
restricted = False,

module_type = None # No Menu

)),
("sync", Storage(
name_nice = T("Synchronization"),
#description = "Synchronization",
restricted = True,

access = "|1|", # Only Administrators can see this module in the default menu & access the controller
module_type = None # This item is handled separately for the menu

)),
("translate", Storage(
name_nice = T("Translation Functionality"),
#description = "Selective translation of strings based on module.",
```

```
module_type = None,

)),
("gis", Storage(
name_nice = T("Map"),
#description = "Situation Awareness & Geospatial Analysis",
restricted = True,
module_type = 1, # 6th item in the menu

)),
("pr", Storage(
name_nice = T("Person Registry"),
#description = "Central point to record details on People",
restricted = True,
access = "|1|", # Only Administrators can see this module in the default menu (access to controller is possible
to all still)
module_type = 10

)),
("org", Storage(
name_nice = T("Organizations"),
#description = 'Lists "who is doing what & where". Allows relief agencies to coordinate their activities',
restricted = True,
module_type = 10

)),
# All modules below here should be possible to disable safely
("hrm", Storage(
name_nice = T("Staff"),
#description = "Human Resources Management",
restricted = True,
module_type = 10,

)),
("vol", Storage(
name_nice = T("Volunteers"),
#description = "Human Resources Management",
restricted = True,
module_type = 10,
```



```
)),
("doc", Storage(
  name_nice = T("Documents"),
  #description = "A library of digital resources, such as photos, documents and reports",
  restricted = True,
  module_type = 10,
)),
("msg", Storage(
  name_nice = T("Messaging"),
  #description = "Sends & Receives Alerts via Email & SMS",
  restricted = True,
  # The user-visible functionality of this module isn't normally required. Rather it's main purpose is to be
  # accessed from other modules.
  module_type = 2,
)),
("cr", Storage(
  name_nice = T("Shelters"),
  #description = "Tracks the location, capacity and breakdown of victims in Shelters",
  restricted = True,
  module_type = 10
)),
("evr", Storage(
  name_nice = T("Evacuees"),
  #description = "Evacuees Registry",
  restricted = True, # use Access Control Lists to see this module
  module_type = 7
)),
("event", Storage(
  name_nice = T("Events"),
  #description = "Activate Events (e.g. from Scenario templates) for allocation of appropriate Resources (Human,
  Assets & Facilities).",
  restricted = True,
  module_type = 10,
)),
]
```

APPENDIX C – Document References

This is a list of links to useful web sites with documentation:

- Development Best Practices:
<http://wiki.sahanafoundation.org/doku.php/deployments:start>
- Sahana Eden website: <http://eden.sahanafoundation.org/>
- Sahana Eden source code: <https://github.com/flavour/eden>

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Apple is a registered trademark of Apple Inc Systems, in the United States, other countries or both.

Google is a registered trademark of Google Inc Systems, in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.