DELIVERABLE REPORT D8.2.2 "Geosocial Mobility and Communication Model"

collaborative project

MASELTOV
Mobile Assistance for Social Inclusion and Empowerment of Immigrants with Persuasive Learning Technologies and Social Network Services

Grant Agreement No. 288587 / ICT for Inclusion

project co-funded by the
European Commission
Information Society and Media Directorate-General
Information and Communication Technologies
Seventh Framework Programme (2007-2013)

Due date of deliverable:	September 30, 2014 (month 33)
Actual submission date:	October 8, 2014
Start date of project:	January 1, 2012
Duration:	39 months

Work package	**WP 8 – WP Community Building Services**
Task	**Task 8.2 – Geosocial Network Services**
Lead contractor for this deliverable	**TI**
Editor	**Massimo Cappello (TI)**
Authors	**Massimo Cappello (TI)**
Quality reviewer	**Ian Dunvell (COV), Lucas Paletta (JR)**

Project co-funded by the European Commission within the Seventh Framework Programme (2007–2013)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	X

VERSION HISTORY

version	date	author	reason for modification	status
001	31.08.2014	Massimo Cappello	First draft version	Internal
002	15.09.2014	Nicoletta Bersia	Internal revision	Internal
003	29.09.2014	Lucas Paletta	Quality review	Internal
004	29.09.2014	Ian Dunvell	Quality review	Internal
005	30.09.2014	Nicoletta Bersia	Final internal revision	Internal

MASELTOV partner			organisation name	country code
01	JR		JOANNEUM RESEARCH FORSCHUNGSGESELLSCHAFT MBH	AT
02	CUR		CURE CENTRUM FUR DIE UNTERSUCHUNG UND REALISIERUNG ENDBENUTZER-ORIENTIERTER INTERAKTIVER SYSTEME	AT
03	AIT		RESEARCH AND EDUCATION LABORATORY IN INFORMATION TECHNOLOGIES	EL
04	UOC		FUNDACIO PER A LA UNIVERSITAT OBERTA DE CATALUNYA	ES
05	OU		THE OPEN UNIVERSITY	UK
06	COV		COVENTRY UNIVERSITY	UK
07	CTU		CESKE VYSOKE UCENI TECHNICKE V PRAZE	CZ
08	FHJ		FH JOANNEUM GESELLSCHAFT M.B.H.	AT
09	TI		TELECOM ITALIA S.p.A	IT
10	FLU		FLUIDTIME DATA SERVICES GMBH	AT
11	BUS		BUSUU ONLINE S.L	ES
12	FUN		FUNDACION DESARROLLO SOSTENIDO	ES
13	DAN		VEREIN DANAIDA	AT
14	MRC		THE MIGRANTS' RESOURCE CENTRE	UK
15	PP		PEARSON PUBLISHING	UK
16	ATE		AUSTRIAN INSTITUTE OF TECHNOLOGY	AT

CONTENT

1. EXECUTIVE SUMMARY

This deliverable describes the work carried out in the scope of task 8.2.2 “Geosocial Network Services” of work package 8 “Community Building Services”.

The result of this task is a software platform with a client component running on Android devices and integrated within the MASELTOV application, and a server component running on a separate infrastructure.

The client component “Help Radar” runs as a service on the smartphone with the goal to detect, notify and localize "buddy proximity" on maps. The purpose of this service is to increase the mobile user context awareness, allowing user-to-user non-intrusive communications when registered users need assistance. The Help Radar will display potential volunteers nearby to the requesting participant. User participation will be on a voluntary basis, and each participant will be able to allow or deny his/her localization.

The volunteer registration process is a critical issue, because it’s important to make sure volunteers are properly accredited. So, in this version a secure registration process has been introduced: by adding a “Validation code” in the User Profile, only well-known users (for example cultural intermediaries) will be able to register themselves as volunteers.

The server component “Help Radar Platform” runs as a service for the client component “Help Radar”. The main goal of this component is to calculate volunteers proximity when registered users ask for assistance, keeping track of users position as well as every user received assistance info.

In this version, Help Radar is able to communicate with the “User Profiling & Recommendation” component in order to obtain and update volunteers profile (for example volunteer rating and knowledge) and also to send some events (for example when user ask for assistance) or to add some coins to the user “wallet” (for example when user rate an assistance).

The previous version of Help Radar service has been evaluated during the first field trials phase (see D9.3 “First Field Trials) and have been consequently updated, The current final version contains the enhancement requested.
In the final version particular attention has been devoted to the ethical issues, introducing the “Validation code” for volunteers. Ethics issues are dealt in “Appendix A: Ethics code of practice” of D.1.1.4 .

The overall MASELTOV system architecture, system specification and services descriptions can be found in D3.1.2, D3.2.2 , D3.3.3 and D6.1.2. These documents are the guidelines for the implementation of all the services.

2. OVERALL ARCHITECTURE

This chapter describes the overall architecture of Help Radar localization platform, which is depictured in Figure 1.

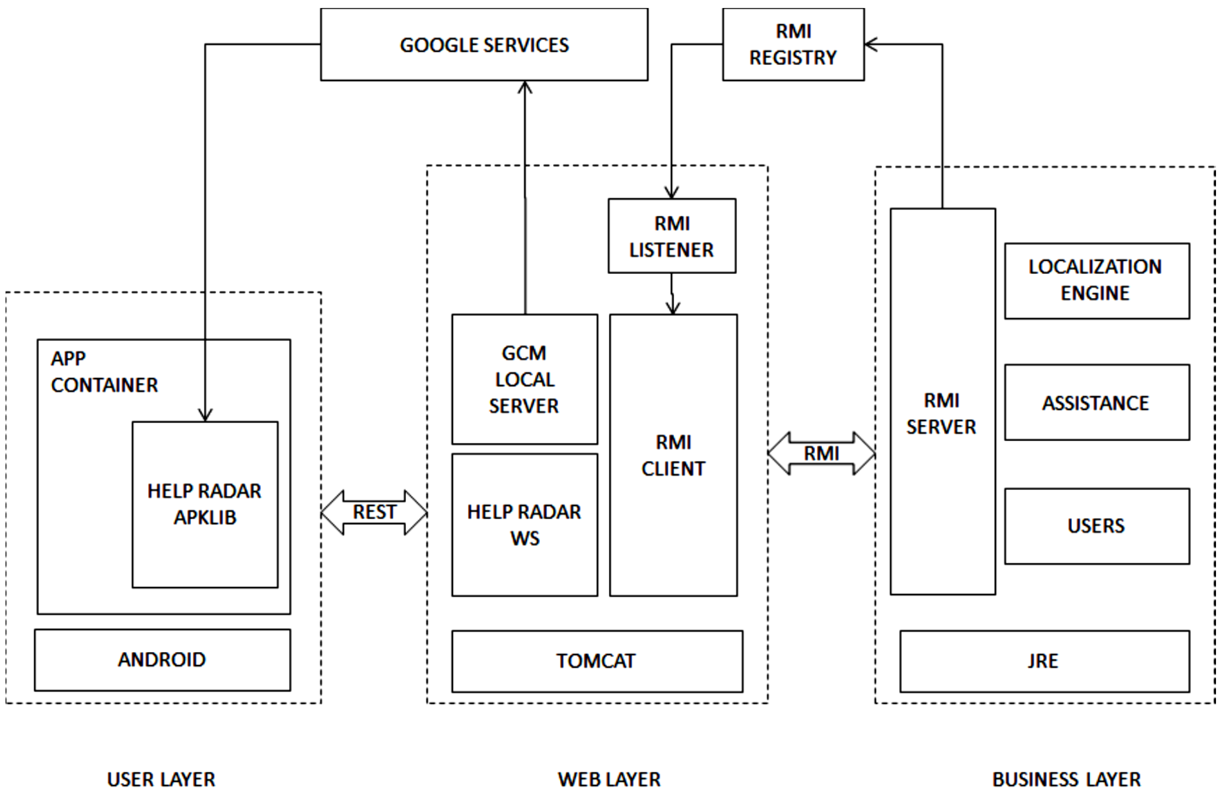


Figure 1: Overall architecture of Help Radar localization platform

Help Radar is a three-layer platform, a User layer on client side, a Web layer and a Business layer on server side. These three layers communicate to each other via two application interfaces, a RestFul interface from Android device to Web layer, and a RMI interface from Web layer to Business layer.

Some external services are used to perform specific functionalities. These services are included in the Google Services infrastructure.

<u>User layer:</u>
As illustrated in [1], Help Radar service is part of the MASELTOV application, a higher level application, that integrates a set of services developed by different project partners. To make service integration as easy as possible, each component of the MASELTOV application has been developed as an APKLIB Android artifact, as is the Help Radar client, connected to the server side services via a RestFul Web Service interface. It also is able to receive messages from Google Cloud Messaging server.

Web layer:

The Web Server layer is in charge of listening at Android device requests. It's composed of one functional component and one infrastructure component:

- ✓ "Help Radar Rest Server : this acts as Web Service server, so it manages all incoming request from Help Radar client running on Android device, and activates the related business components via the RMI interface
- ✓ "RMI client": it activates services on Business layer

Business layer:

This layer implements the localization platform business logic. It's composed by four functional components and one infrastructure component:

- ✓ "GCM local server": it's part of Google Cloud Messaging infrastructure (for a full description of this service see [5]), it receives incoming messages from Android device and forwards them to CGM Server (part of "Google Services")
- ✓ "Localization engine": it searches volunteers near the users, performing distance calculation from user and all users registered as volunteers.
- ✓ "Assistance": it manages the assistance requests, recording the messages exchange from user and volunteers and calculating volunteers rate
- ✓ "Users": it manages user subscription to Help Radar service and volunteer sign-up
- ✓ "RMI server": it manages RMI interface, recording Help Radar services on RMI Registry and waiting for incoming request from Web Server side. Then, it activates business services available on the layer.

3. CLIENT SIDE

This chapter describes the Help Radar APKLIB software components and functionality. Starting from a description of the client side architecture, a complete description of Help Radar functionality and GUI is provided, and the main application data managed by Help Radar listed.

In order to better understand the internal software components, two of most important Android app configuration and build files (AndroidManifest.xml and pom.xml) are described.

3.1 ARCHITECTURE

The Help Radar APKLIB is composed of a set of application components that implement the Help Radar functionality. These components are active only when the Help Radar service is running. However, there are also some application services (internal services) that are activated at the first Help Radar startup and remain active even if the application is not running or it runs in background. These services perform some background activities, such us listening at incoming messages, or device location.

Other non-Help Radar services (external services) are also used to implement some specific functionalities. In particular, the volunteer contact message exchange is based on Google Cloud Messaging service.

Figure 2 depicts the Help Radar APKLIB architecture:

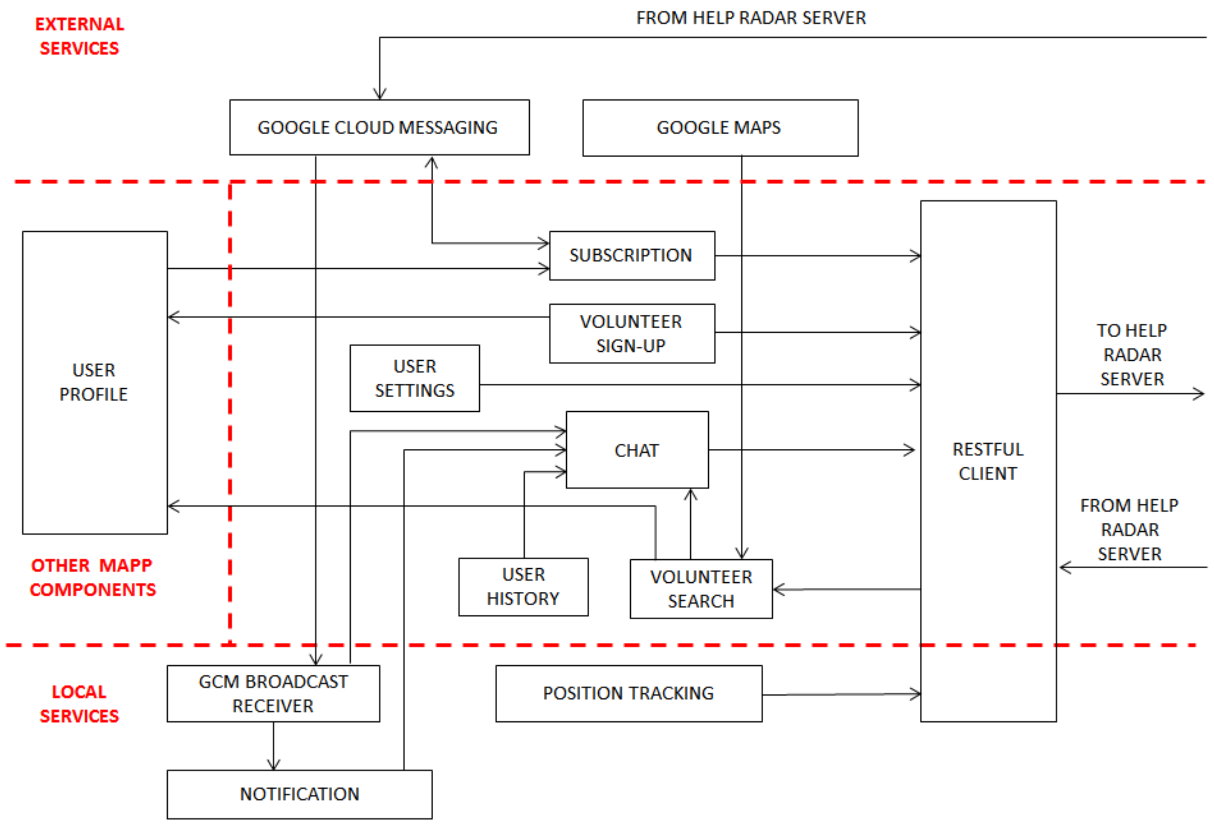


Figure 2: Client side architecture of APKLIB architecture

3.1.1 COMPONENTS DESCRIPTION

This section describes the Help Radar components and services, as depicted in Figure 2.

Table 1 – Help Radar components and services

Name	Description
Subscription	This component is responsible for the user registration on the “Google Cloud Messaging (GCM)” server, and the user subscription on Help Radar local server. It’s activated during the first Help Radar run. It interacts with Help Radar server via a RestFul connection and with GCM via an HTTP connection. It interacts with User Profile to get the user login parameters (eg nickname, uinqueID,..).
Volunteer search	This component manages the whole volunteer search wizard, and displays on list/map the volunteers found near the user. To display users’ information, it interacts with “Google Maps” service. It also starts the volunteer contact process, via a Chat component. It interacts with User Profile to send some events.
User history	This component is to show the list of the requested by the user, loading it from the Help Radar server as well as all the related messages. It also manages the assistant rating. It can interact with

	the "Chat" component when the user wants to continue a dialog with the volunteer without re-starting the search process.
Chat	This component is responsible for the user-to-volunteer communication starting after the volunteer search process. It's able to send messages to the volunteer and display the whole message exchange. It can be activated from some app components (Volunteer search and User History) when the app is running, but also from the Android notification service (via "GCMBroadcast Receiver") when the app is not running.
User settings	This component manages the user preferences, saving them as shared preferences. It can perform backup/restore operations if the user needs to save his/her preferences for security reasons.
Volunteer signup	This component manages the volunteer signup process, saving the volunteer profile into Help Radar server database. It also starts the user position tracking activity. It interacts with User Profile to send some events.
Position tracking	This service localizes the user's position and sends location data to Help Radar server. If the user is registered as volunteer, this service sends location data periodically and remains active even when the app is not running. If the user is not a volunteer, this service is activated during the search process only.
GCM Broadcast receiver	This service waits for incoming messages from "Google Cloud Messaging" server and wakes up the "Chat" component to display the message and continue the dialog with user/volunteer. If the app is not running, (or the Chat page is not opened), the component produces an Android notify.
Notification	The Android notification service is used to warn the volunteer of a new incoming message from a user, when the Help Radar app is not running or the Chat page is not open. If the volunteer opens the notification message, the Chat page is opened and the volunteer is able to reply to user.
RestFul client	This component manages interactions between the Android device and the Help Radar server via REST protocol.
Google Maps	This external service is used by "Volunteer search" component to display volunteers on geographic map
Google Cloud Messaging	This external service is used to manage the message exchange between the user and the volunteer.

3.1.2 PACKAGES

This paragraph describes shortly the Help Radar software packages.

Package's name	**Description**
com.tilab.georadar	It contains the main application Activity classes
com.tilab.georadar.connection	It contains asynchronous Task classes used to communicate with Help Radar server

com.tilab.georadar.customs	It contains all UI Android customization classes
com.tilab.georadar.map	It contains classes for managing visualization of volunteers on maps
com.tilab.georadar.util	It contains Java utility classes
com.tilab.geocontainer	It contains classes for managing the message exchange between user and volunteer, including interactions with Google Cloud Messaging service

3.1.3 ANDROID MANIFEST.XML

The following paragraphs describe information contained in Help Radar AndroidManifest.xml file.

3.1.3.1 GENERAL

Name	Description
Package	*com.tilab.georadar*
versionCode	*7*
versionName	*0.0.7-SNAPSHOT*
minSdkVersion	*15*
targetSdkVersion	*16*
theme	*Theme.Holo.Light*

3.1.3.2 PERMISSIONS

Name	Description
android.permission.ACCESS_COARSE_LOCATION	Allows Help Radar to access approximate location derived from network location sources such as cell towers and Wi-Fi.
android.permission.ACCESS_FINE_LOCATION	Allows Help Radar to access precise location from location sources such as GPS, cell towers, and Wi-Fi.
android.permission.ACCESS_NETWORK_STATE	Allows Help Radar

	to access information about networks
android.permission.INTERNET	Allows Help Radar to open network sockets.
android.permission.GET_ACCOUNTS	Allows access to the list of accounts in the Accounts Service
android.permission.WAKE_LOCK	Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming
android.permission.VIBRATE	Allows access to the smartphone vibrator
com.tilab.georadar.permission.C2D_MESSAGE	With this permission only Help Radar can receive data messages from GCM service
com.google.android.c2dm.permission.RECEIVE	Allows GCM Broadcast Receiver to receive message from GCM Server
com.google.android.c2dm.permission.SEND	With this permission only GCM service can send data messages to Help Radar
com.tilab.georadar.permission.MAPS_RECEIVE *android.permission.WRITE_EXTERNAL_STORAGE* *com.google.android.providers.gsf.permission.READ_GSERVICES*	Allow access to Google Map v2
com.ait.userprofile.AITUserProfileProvider	Allow access to User Profile

3.1.3.3 LIBRARIES

Name	**Description**
com.google.android.maps	Used to display volunteers on geographic maps

3.1.3.4 ACTIVITIES

Name	Description
`com.tilab.georadar.GeoRadarActivity`	Main Help Radar Activity. Implements Help Radar dashboard and "Subscription" component. It's activated from MASELTOV app dashboard with "com.tilab.GEORADAR" name
`com.tilab.geocontainer.ChatActivity`	Implements part of "Chat" component
`com.tilab.georadar.AssistanceHistoryActivity`	Implements part of "User history" component
`com.tilab.georadar.VolunteerListActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.SettingsActivity`	Implements "User settings" component
`com.tilab.georadar.KnowledgeSelectionActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.VolunteerDetailActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.VolunteerHistoryActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.VolunteerSignupActivity`	Implements "Volunteer sign-up" component
`com.tilab.georadar.LanguageSelectionActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.TimetableActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.VolunteerMapActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.VolunteersTabsActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.VolunteerHistoryTabsActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.LanguageMultiSelectActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.HistoryTabActivity`	Implements part of "User history" component
`com.tilab.georadar.VolunteerOpenHistoryActivity`	Implements part of "Volunteer search" component
`com.tilab.georadar.AssistanceRatingActivity`	Implements part of "Chat" component
`com.tilab.geocontainer.ChatHistoryActivity`	Implements part of "Chat"

	component

3.1.3.5 Services and Receivers

Name	Description
com.google.android.gcm.GCMBroadcastReceiver	Implements part of "GCM Broadcast Receiver" service
com.tilab.geocontainer.GCMIntentService	Implements part of "GCM Broadcast Receiver" service
com.tilab.georadar .connection.VolunteerLocalizationService	Implements part of "GCM Broadcast Receiver" service
com.tilab.georadar.connection.ServicesManager	Starts up the Help Radar services and receivers when user logs in
com.tilab.georadar.connection.GCMPingService	Implements part of "GCM Broadcast Receiver" service
com.tilab.georadar.connection.LocalizationBroadcastReceiver	Implements part of "Position Tracking" service
com.tilab.georadar.util.BootReceiver	Starts up the Help Radar services and receivers when device boots
com.tilab.georadar.util.LoggingReceiver	Stops all Help Radar services when user logs out

3.1.3.6 Google Maps KEY

Name	Value
com.google.android.maps.v2.API_KEY	*AIzaSyC5JgMx4audnHppDX6gZ_yuOeiRuZc3wsA*

3.1.4 POM.XML

The following paragraphs describe information contained in Help Radar pom.xml file.

3.1.4.1 GENERAL

Name	Description
groupId	com.tilab
artifactId	georadar
packaging	apklib
version	0.0.10-SNAPSHOT

3.1.4.2 REPOSITORIES

Name	Description
http://repo.maven.apache.org/maven2	Central maven repository
http://ooo-maven.googlecode.com/hg/repository	
http://sesamo.tilab.com/releases	TI public repository
`http://maven.fluidtime.com/content/repositories/public-maseltov`	Fluidtime public repository

3.1.4.3 DEPENDECIES

Name	Description
com.google.android 4.1,1,4	
com.google.android support-v4 r7	
com.google.android gcm-client r3	
com.google.code.gson 2.2.2	
org.apache.httpcomponents httpclient 4.2.5	
org.apache.httpcomponents httpmine 4.2.5	
org.apache.httpcomponents httpclient-cache 4.2.5	
org.apache.httpcomponents fluent-hc 4.2.5	
com.google.android.gms google-play-services 12	
com.google.android.gms google-play-services 12	

3.1.5 SHARED PREFERENCES

This following paragraph describes the Android shared preferences managed by Help Radar. Some preferences are used to keep track of the current user or volunteer status, and some preferences reflect the application settings, detailed in par. 3.2.

Name	Description

Unique_id	MASELTOV unique user identifier
id of user language	user language
userId	local Help Radar user identifier
Registered	flag indicating the user subscription at Help Radar service
isVolunteer	flag indicating the user registration as volunteer
isActive	volunteer status (hide, active)
displayMode	see par. 3.2
languagePreference	user profile language
searchArea	see par. 3.2

3.2 APPLICATION SETTINGS

Help Radar comes with a set of default parameters. These parameters are used to manage some specific application behaviours, they are user specific, so each user can modify them through the "Settings" functionality.

Application settings are saved locally into the Android device as shared preferences.

Application settings managed by Help Radar are the following:

1) default display mode: this parameter sets the default volunteer visualization mode after a search in the "Request for Assistance" function. Available values: List, Map. Default value: List
2) default language for assistance: this parameter lets the user select the language of volunteers he/she is looking for. Available values: My default language, Don't care, Always ask. If user selects "Always ask", then the volunteer language selection form will be displayed in the volunteer search wizard, if user selects "My default language" then only volunteers speaking the same user language will be searched, if user selects "Don't care" then no language filter will be applied in volunteer search. Default value: My default language.
3) search radius: this parameter sets the radius of the area around the user in which Help Radar will be active. Available values: from 500mt, to 100 km. Default value: 20 km.

3.3 FUNCTIONAL DESCRIPTION

The following paragraphs describe information contained in Help Radar AndroidManifest.xml file.

3.3.1 HELP RADAR STARTUP AND SUBSCRIPTION

Whenever Help Radar is launched from Mapp main dashboard, the following user info are read from User Profile :

- nickname
- uniqueID (email)
- language

- gender
- validation code

User uniqueID is checked against the local Help Radar preferences in order to verify if the user is already subscribed to Help Radar service:

- if not, user can start the subscription process;
- if yes, Help Radar will start some preliminary checks and settings: application settings are loaded as well as user local settings. If user is also signed up as volunteer, some other specific volunteer parameters are loaded from the remote server.

Language parameter will be used to set-up the GUI layout.

Validation code parameter will be used to allow user for signin-up as volunteer.

At the end of startup and subscription process, user will have access to the Help Radar dashboard.

3.3.2 USER LOCATION

User location, and dispatch of related data to the remote Help Radar server, will take place in the following cases:

- when user starts a volunteer search phase: localization data is sent in one shot mode
- when user refreshes volunteer list (after a volunteer search): data is sent in one shot mode
- when user signs up him/herself as volunteer: from this time on, location data is sent to Help Radar server periodically (every 30 seconds), until user resigns him/herself from being a volunteer. In other words, location process will run as a background service, then will be active even if Help Radar app (and MASELTOV app) is not running.

3.3.2.1 LOCATION STRATEGY

To determinate the user's geographical position using Android devices it is possible to use a GPS Location Provider, Network Location Provider, or both. GPS Location Provider utilizes GPS to acquire user location, Network Location Provider utilizes WiFi or Cell-id.

Although GPS is most accurate, it only works outdoors, quickly consumes battery power, and doesn't return the location as quickly as users want. Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, and providing localization information both in case of indoors and outdoors situations , it replies faster, and uses less battery power.

Help Radar location service can use both GPS and the Network Location Provider, depending on accuracy level criteria. Although this criteria could accept three accuracy values (high, medium, low), its value is always set on “high”.

Depending on accuracy level criteria (always set on “high”), GPS availability and indoor/outdoor user position, Android Location Provider framework automatically chooses the best Location Provider to use.

As mentioned before, one of main issues in location is the battery power consumption. The Location Provider choice is not enough to determinate the best model, because there is another aspect to be considered that is the frequency of location update. Location update interval can be controlled using two parameters:

- the “minTime” parameter: the elapsed time between location updates will never be less then minTime
- the “minDistance” parameter: location provider will only send to Help Radar an update when the location has changed by at least “minDistance” meters, AND at least after “minTime” milliseconds

Actually, the Help Radar settings are:

- minTime: 30 sec
- minDistance: 0 m.

3.3.3 Request for Assistance

This functionality allows users to find out the presence of volunteers located to a determined distance from him/her, and contact them in a non intrusive way (by chat). By managing a specific parameter setting, user can determine the maximum search distance.

3.3.3.1 Volunteer Search

In the volunteer searching process, the user is driven by a "wizard" which simplifies the search by providing a step-by-step approach:

- Step 1: The user may select one or more interested knowledge areas (e.g. school, law, health,…) . If the user doesn’t select anything, search is performed on the whole list of registered volunteers, and no knowledge filter is applied.
- Step 2: Depending on a specific setting value, user may select one or more of volunteer languages. If this step is disabled, search is performed without applying any language filter.
- Step 3: The search starts on the remote Help Radar server, seeking for volunteers in the area around the user. Located volunteers are displayed on user device and presented on a map as well as on a detailed list.

3.3.3.2 VOLUNTEER DISPLAYED ON A LIST

The list of Volunteers is organised and displayed according to increasing distance from the user who has started the search: the first volunteer is the nearest to the user.

For each volunteer, some general information is presented. Tapping on a specific element of the volunteers‘ list item, a richer and more detailed view is presented. From this view, the user can start the contact process, enabling a new chat session with the selected volunteer. Help Radar chat has been implemented on the Google Cloud Message Infrastructure ([5]). More details on contact process are described in par. 3.2.
.

3.3.3.3 VOLUNTEER DISPLAYED ON THE MAP

Volunteers are displayed on a map covering a geographic area wide enough to contain all the volunteers that match the request. The map is then centered on the current user position.

Tapping on a volunteer bookmark, a bubble with some general information is opened. Tapping again on the bubble a richer and more detailed view is presented. From this view, the user can start the contact process, enabling a new chat session with the selected volunteer. Help Radar chat has been deployed on the Google Cloud Messaging Infrastructure ([5]). More details on this contact process are described in Section 3.3.3.4.

3.3.3.4 VOLUNTEER CONTACT PROCESS

The volunteer contact process has been designed to be less intrusive as possible from the volunteer point of view. Having this goal in mind, users that need assistance may contact volunteers and keep in touch with them by a chat based messaging exchange.

Help Radar chat is based on the Google Cloud Messaging for Android. Basically, GCM is a service that allows you to send data from a server-side application to Android-powered devices. The GCM service handles all aspects of queuing of messages and delivery to the target Android application running on the target device.

To receive messages from GCM, the Android applications must obtain a GCM key first: each device will have a specific GCM key. This key will be used by the server-side application (e.g. a servlet running under Tomcat) to forward messages to a specific device, via GCM server.

In Help Radar application, the Android device obtains its GCM key during the user subscription phase, so that each user profile information, including the key, will be forwarded to Help Radar server with a single Web Service command. At the end of the subscription phase, users will be able to exchange messages with registered volunteers.

Messages coming from GCM are managed from a Help Radar process running as a background service, the CGM Broadcast Receiver. As an Android service, CGM Broadcast Receiver is activated on Help Radar first startup and remains active even if the Help Radar is not running. So, when a new message is received, CGM Broadcast Receiver is able to generate a specific Android notification and wake up the chat process.

To keep alive the connection from Help Radar to GCM server , a specific Help Radar service, GCM Ping Service, has been deployed; this service send to Help Radar a "ping" message via GCM server every 60 seconds

In Figure 3, the process above described is illustrated and detailed. Red items and lines refer to volunteer's process, black items and lines to user's process.

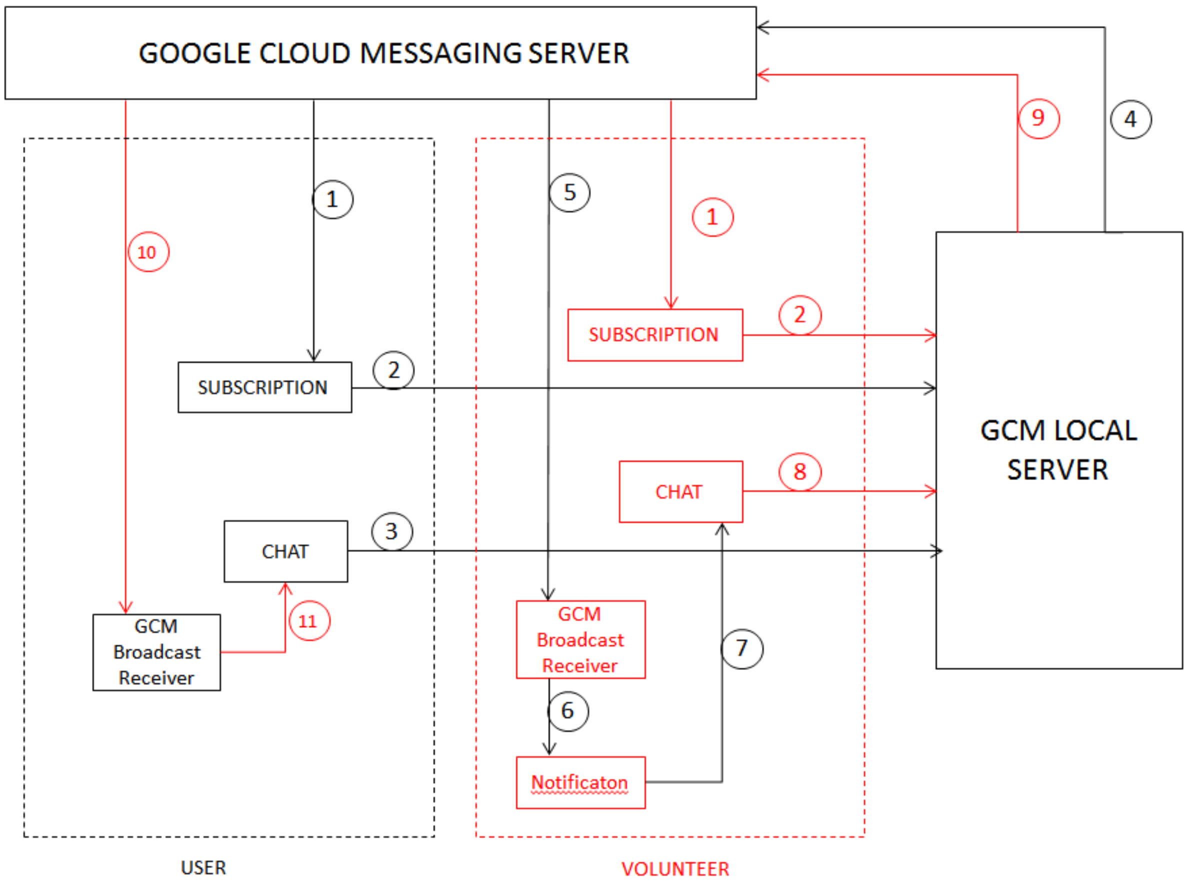


Figure 3: Volunteer contact process

<u>SUBSCRIPTION PHASE</u>

1) User/volunteer mobile devices register theirselves on Google Cloud Message Server and get a valid GCM key
2) Users register theirselves on CGM Local Server (part of Help Radar local server), saving the device GCM key in the user profile

<u>MESSAGES EXCHANGE PHASE</u>

3) USER contact VOLUNTEER writing a message: the message is sent to GCM Local Server
4) GCM Local Server saves the message into Help Radar database and forwards the message to GCM Server. VOLUNTEER is identified by his device GCM key
5) GCM Server delivers the message to the volunteer whose device is identified by GCM key
6) The CGM Braodacast Receiver creates, on the volunteer device, an Android notification that will be displayed on the device status bar
7) VOLUNTEER can read the notification and can open the Help Radar chat page: the message from USER is displayed. VOLUNTEER can reply to USER
8) VOLUNTEER replies to USER writing an answer: : the message is sent to GCM Local Server
9) GCM Local Server saves the message into Help Radar database and forwards the message to GCM Server. USER is identified by his device GCM key
10) GCM Server delivers the message to user device identified by GCM key
11) The CGM Braodacast Receiver on the user device displays the reply on the chat page

3.3.3.4.1 Assistance record

When users start a volunteer contact process, a new assistance record is created into Help Radar database. This record remains open until user rates it. This means that every message exchanged between a specific user-volunteer couple will be added to a particular assistance record until user closes the assistance.request session

A user can have just one assistance session opened with a specific volunteer.

Please note points 4 and 9 of contact process: all messages from/to users are saved into Help Radar database. This behaviour lets user read in every moment (see Assistance History feature in next paragraph) the old messages and get some useful information from the past interactions.

3.3.4 ASSISTANCE HISTORY

This functionality allows user to view the list of the requested assistances and to produce a user satisfaction indicator on received helps, rating each assistance through a vote from 1 star to 5 stars (with a half-star rate available).

The user can also open each assistance record and view the whole history of messages exchanged with the volunteer. Furthermore, if the selected assistance record is still open (not yet rated), the message history is “alive”, that is the user can try to contact the volunteer sending him/her other messages, without the need to restart the volunteer search as described in “Request for Assistance” paragraph.

If the user has signed-up him/herself as volunteer, he/she can also view the list of the provided assistances

3.3.5 SIGN-UP AS VOLUNTEER

This functionality allows the user to declare him/herself as a volunteer. The volunteer registration process is a critical issue, because it's important to make sure volunteers are properly accredited. So, in this version a secure registration process has been introduced: by adding a "Validation code" in the User Profile, only well known users (for example cultural intermediaries) will be able to register themselves as volunteers.
The Validation Code may be provided to user by organizations responsable of the selection of volunteers (eg. NGOs).

The overall validation process is the following:

- User gets the Validation Code from NGOs and fills the appropriate field of User Profile (the VALIDATION CODE field) with this code;
- Help Radar maintains the list of authorized Validation Codes into the Help Radar database
- Help Radar reads the code from the User profile during the start-up phase and checks the code against its list
- If the code is valid the "Sign-up as volunteer" functionality will be enabled

Now the user can set the necessary volunteer profile data:

- Knowledge: user must select on or more competences form which he declares him/herself as an expert
- Languages: user must select one or more languages in which he/she is able to give assistance
- Available daytime: user must select the days of week, and the hours of day in which he/she is available to give assistance.

After that, the volunteer profile data are sent to Help Radar server.
From now on, the user is recognized as volunteer, the localization tracking service is activated on the user device, and the volunteer position data (see "User Localization" paragraph.) are periodically sent to the Help Radar server.

User can remove his/her registration as a volunteer at any time.

User can temporarily suspend his availability at any time, hiding his position, that is his position data will not be sent to Help Radar server until the volunteer declares him/her as available again.

3.3.6 SETTINGS

This functionality allows the user to set, or update his/her application settings. These parameters are saved into Android device, but also into the Help Radar server database. In this way it's possible to restore them at any time.

3.3.7 SENDING DATA TO USER PROFILE

Interaction between Help Radar and User Profile is not limited to getting data in the start-up and subscription process (see par. 3.3.1). In some other circumstances Help Radar sends data to User Profile. These data may be:

- events
- coins

3.3.7.1 SENDING EVENTS TO USER PROFILE

Help Radar sends an event to User Profile when:

- user contacts a volunteer to start a chat communication (CONTACT event)
- user rates a received assistance (RATINGASSIST event)
- user sign-up him/herself as volunteer (SIGNUPVOL event)
- user remove him/herself from volunteer (REMOVEVOL event)
- volunteer change his/her availability status (CHANGESTATUS event)

3.3.7.1.1 CONTACT event

Events parameters sent to UserProfile are described in the following table:

Parameter name	Description
username	nickname of the user starting the chat with the volunteer
volunteerUsername	nickname of the contacted volunteer
reqKnowledges	list of knowledges of the contacted volunteer
reqLanguage	list of languages of the contacted volunteer

3.3.7.1.2 RATINGASSIST event

Events parameters sent to UserProfile are described in the following table:

Parameter name	Description
username	nickname of the user rating the volunteer
volunteerUsername	nickname of the rated volunteer
rating	assistance rate

3.3.7.1.3 SIGNUPVOL event

Events parameters sent to UserProfile are described in the following table:

Parameter name	Description
username	nickname of the user that wants to sign-up him/herself as volunteer
knowledges	list of knowledges of the volunteer
languages	list of languages of the volunteer

status	availability status of the volunteer

3.3.7.1.4 REMOVEVOL event

Events parameters sent to UserProfile are described in the following table:

Parameter name	Description
volunteerUsername	nickname of the user that wants to remove him/herself from volunteer

3.3.7.1.5 CHANGESTATUS event

Events parameters sent to UserProfile are described in the following table:

Parameter name	Description
volunteerUsername	nickname of the contacted volunteer
status	new availability status of the volunteer

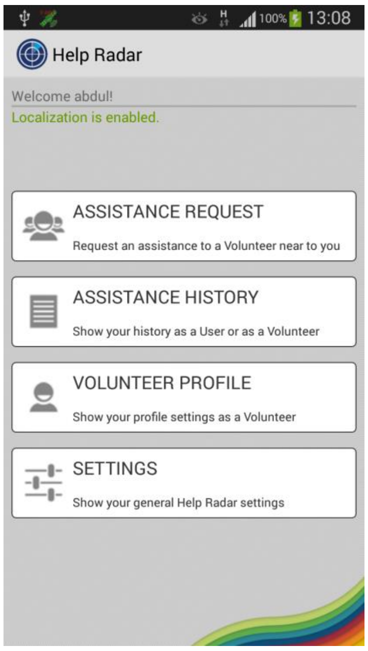
3.3.7.2 SENDING COINS TO USER PROFILE

Whenever user rates a received assistance, Help Radar adds 50 coins to the user’s “wallet”.

3.4 USER INTERFACE

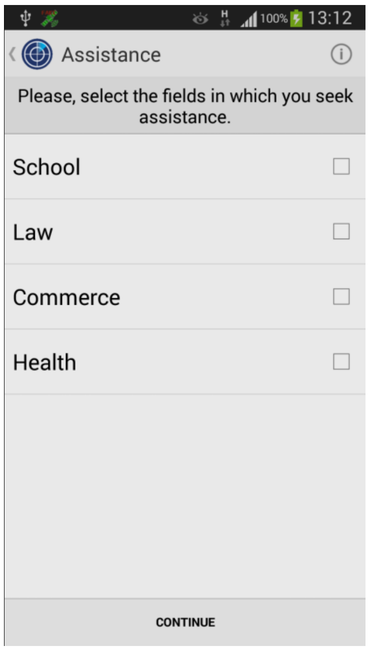
The Help Radar UI is available (professional translations) in Italian, English, Spanish, Turkish and Arabic.
The Help Radar dashboard is reached starting from the general MASELTOV dashboard, which contains icons accessing all possible MASELTOV services, and tapping on the "Help Radar" icon.

3.4.1 HELP RADAR DASHBOARD

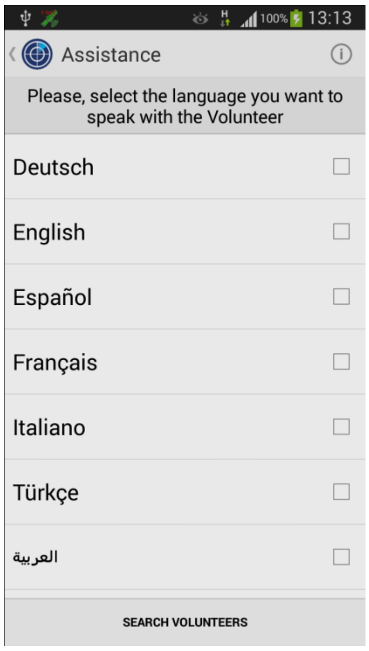
 13:08 Help Radar Welcome abdul! Localization is enabled. ASSISTANCE REQUEST Request an assistance to a Volunteer near to you ASSISTANCE HISTORY Show your history as a User or as a Volunteer VOLUNTEER PROFILE Show your profile settings as a Volunteer SETTINGS Show your general Help Radar settings	• The user can select one function by tapping it

3.4.2 Assistance Request

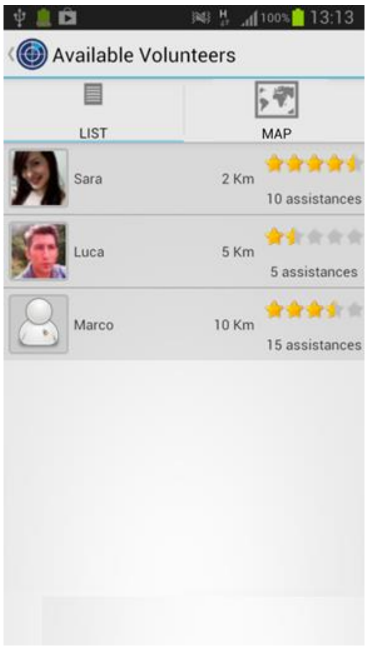
3.4.2.1 Competences Selection

 13:12 Assistance. Please, select the fields in which you seek assistance. School ☐ Law ☐ Commerce ☐ Health ☐ CONTINUE	• The user can check one or more selections according to the competences needed (one or more) by tapping the selection • He/she can also select nothing (it means that any competence is ok). • At the end he/she has to tap “Continue”.

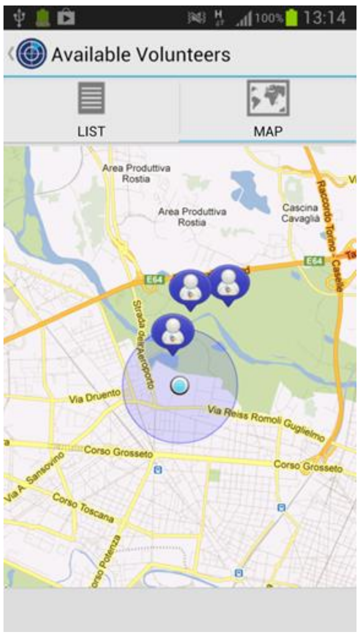
3.4.2.2 Language Selection

 13:13 Assistance. Please, select the language you want to speak with the Volunteer. Deutsch ☐ English ☐ Español ☐ Français ☐ Italiano ☐ Türkçe ☐ العربية ☐ SEARCH VOLUNTEERS	• The user may select the required volunteer’s language tapping it. This screen visualization depends on ‘default language for assistance’ setting. • Then he/she has to tap on “Search volunteers”. • The Help Radar Platform starts searching for available volunteers according to the above selected features and the distance settings as illustrated in 3.4.5.4 .

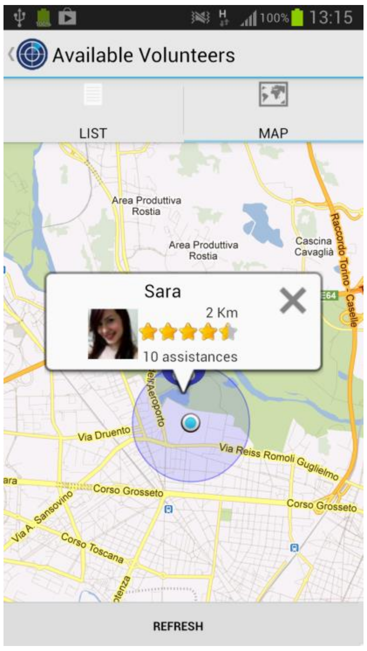
3.4.2.3 AVAILABLE VOLUNTEERS LIST

 Available Volunteers / LIST / MAP / Sara 2 Km 10 assistances / Luca 5 Km 5 assistances / Marco 10 Km 15 assistances	• The available volunteers can be shown in a **list** as well as on a **map**. • The **list** is ordered by distance (the nearest volunteer on top). • For each volunteer the following information are shown: volunteer photo, if available, his/her current distance, the summary of given assistances (number and rating) • Tap on an element list to display more volunteer details

3.4.2.4 AVAILABLE VOLUNTEERS MAP

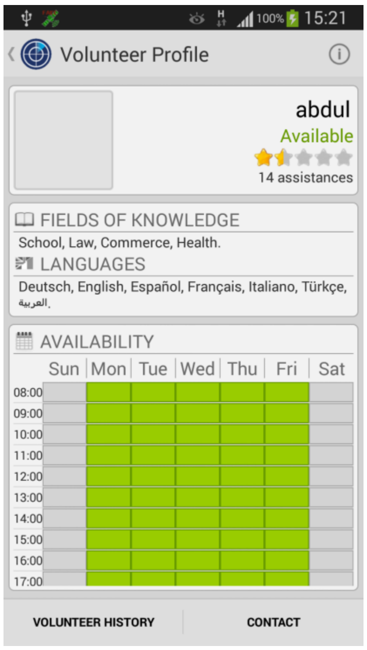
	• If the user taps a volunteer picture in the **map**, a callout window is displayed with: • some information on the selected volunteer: • an image (photo or what else he has chosen) • his/her current distance from the user • the summary of given assistances (number and global rating)

3.4.2.5 VOLUNTEER SELECTION ON MAP



- Tap on the callout window to display volunteer details
- Tap on X to close the callout window

3.4.2.6 VOLUNTEER DETAILS



abdul
Available
14 assistances

FIELDS OF KNOWLEDGE
School, Law, Commerce, Health.

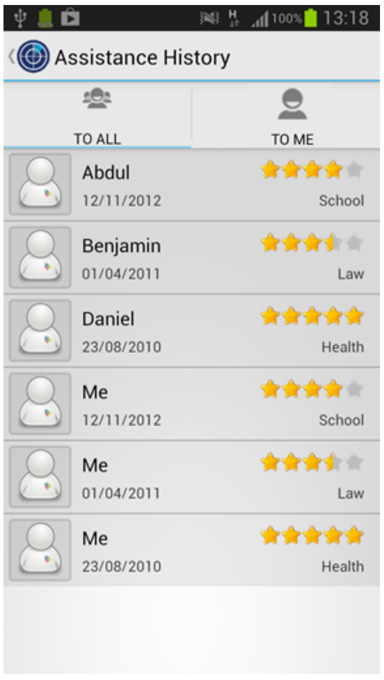
LANGUAGES
Deutsch, English, Español, Français, Italiano, Türkçe, العربية.

AVAILABILITY

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
08:00	▨	🟩	🟩	🟩	🟩	🟩	▨
09:00	▨	🟩	🟩	🟩	🟩	🟩	▨
10:00	▨	🟩	🟩	🟩	🟩	🟩	▨
11:00	▨	🟩	🟩	🟩	🟩	🟩	▨
12:00	▨	🟩	🟩	🟩	🟩	🟩	▨
13:00	▨	🟩	🟩	🟩	🟩	🟩	▨
14:00	▨	🟩	🟩	🟩	🟩	🟩	▨
15:00	▨	🟩	🟩	🟩	🟩	🟩	▨
16:00	▨	🟩	🟩	🟩	🟩	🟩	▨
17:00	▨	🟩	🟩	🟩	🟩	🟩	▨

- The following Extended volunteer information are displayed:
  - ✓ Photo
  - ✓ Number of given assistances
  - ✓ Global rating
  - ✓ Competences
  - ✓ Language spoken
  - ✓ daytime Availability
- Tap on Volunteer History button to display the volunteer assistance list
- Tap on “Contact” button to start a chat session with the selected volunteer.

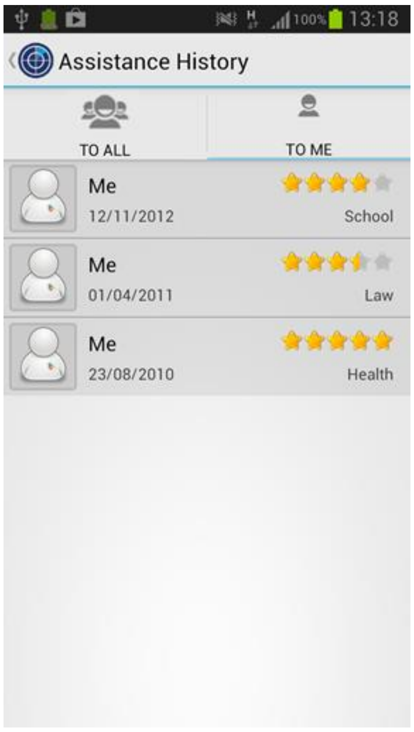
3.4.2.7 ALL VOLUNTEERS HISTORY



Abdul	12/11/2012	★★★★☆ School
Benjamin	01/04/2011	★★★½☆ Law
Daniel	23/08/2010	★★★★★ Health
Me	12/11/2012	★★★★☆ School
Me	01/04/2011	★★★½☆ Law
Me	23/08/2010	★★★★★ Health

- The “Assistance History” shows the user the details related to the selected volunteer such as:
- Date of assistances provided
- Field of assistance (see competencies)
- Nickname of the assisted person
- rate of the assistance
- Tapping on “To all” TAB the assistances given by the selected volunteer to all users are displayed.

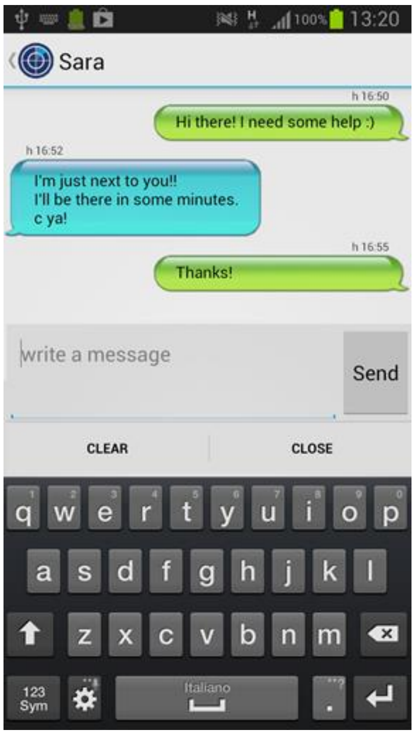
3.4.2.8 PERSONAL VOLUNTEER HISTORY



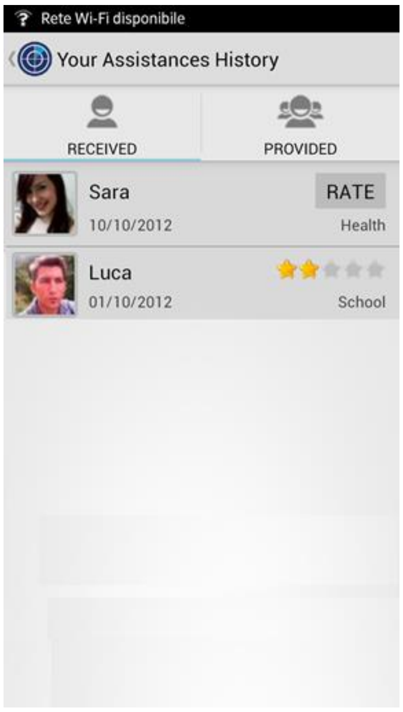
Me	12/11/2012	★★★★☆ School
Me	01/04/2011	★★★½☆ Law
Me	23/08/2010	★★★★★ Health

- The “Assistance History” shows the user the details related to the selected volunteer such as:
- Date of assistance
- Field of assistance (see competencies)
- Nickname of the assisted person
- rate of the assistance
- Tapping on ‘To me’ TAB assistances already received by the user are displayed

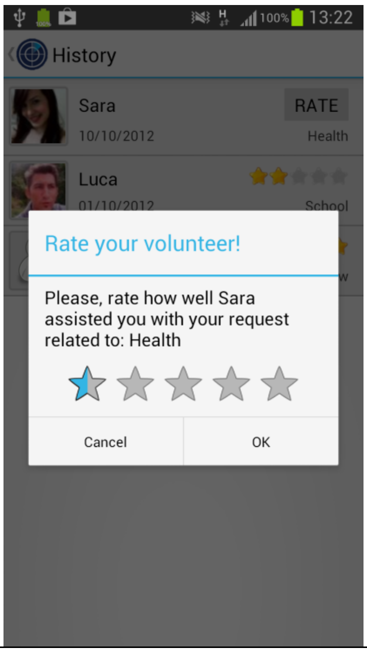
3.4.2.9 CONTACT VOLUNTEER

 Sara h 16:50 Hi there! I need some help :) h 16:52 I'm just next to you!! I'll be there in some minutes. c ya! h 16:55 Thanks!	• The selected volunteer can be contacted by starting a chat session • Tap on Close button to finish the chat session

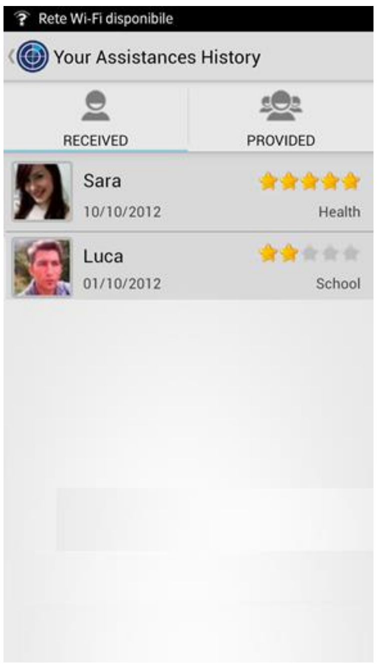
3.4.3 ASSISTANCE HISTORY

 Rete Wi-Fi disponibile Your Assistances History RECEIVED | PROVIDED Sara — 10/10/2012 — RATE — Health Luca — 01/10/2012 — School	Shown details are: • Date of assistance • Field of assistance (see competencies) • Nickname of the volunteer who gave the assistance • Rate of the received assistance or button to rate the received assistance

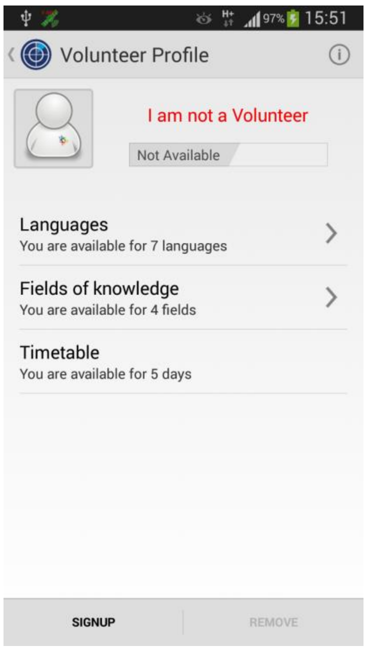
3.4.3.1 RATE AN ASSISTANCE

 100% 13:22 History Sara — 10/10/2012 — RATE — Health Luca — 01/10/2012 — School Rate your volunteer! Please, rate how well Sara assisted you with your request related to: Health Cancel | OK	Rate the received assistance by tapping on stars

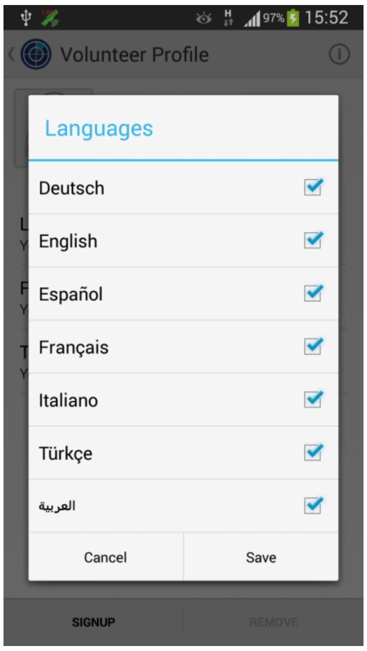
3.4.3.2 ALL ASSISTANCES RATED

 Rete Wi-Fi disponibile Your Assistances History RECEIVED PROVIDED Sara 10/10/2012 ★★★★★ Health Luca 01/10/2012 ★★☆☆☆ School	Shown details are: • Date of assistance • Field of assistance (see competencies) • Nickname of the volunteer who gave the assistance • Rate of the received assistance

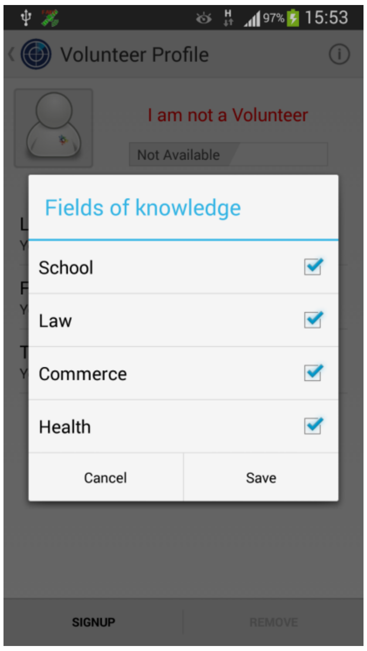
3.4.4 VOLUNTEER PROFILE

 15:51 Volunteer Profile I am not a Volunteer Not Available Languages You are available for 7 languages Fields of knowledge You are available for 4 fields Timetable You are available for 5 days SIGNUP REMOVE	The user can set his/her volunteer profile, if he/she wants to give assistance to other users The user taps on Signup button to declare him/herself as a volunteer

3.4.4.1 LANGUAGE SELECTION

 15:52 Volunteer Profile Languages Deutsch ✓ English ✓ Español ✓ Français ✓ Italiano ✓ Türkçe ✓ العربية ✓ Cancel Save SIGNUP REMOVE	If the user taps Languages (see 3.4.4 smartphone screenshot) he/she can select the languages he/she is able to speak

3.4.4.2 COMPETENCES SELECTION

	If the user taps Fields of knowledge (see 3.4.4 smartphone screenshot he/she can select the competences he/she has

Volunteer Profile

I am not a Volunteer

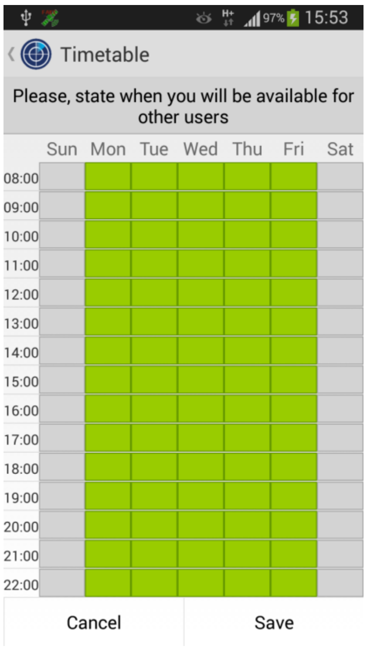
Not Available

Fields of knowledge

School	✓
Law	✓
Commerce	✓
Health	✓

Cancel Save

3.4.4.3 AVAILABILITY TIMETABLE

	If the user taps Timetable (see 3.4.4 smartphone screenshot) he/she can select the days and hours in which he/she may be available

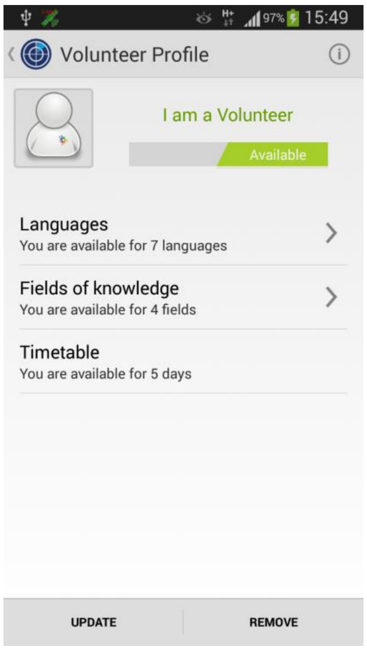
Timetable

Please, state when you will be available for other users

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
08:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
09:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
10:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
11:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
12:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
13:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
14:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
15:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
16:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
17:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
18:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
19:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
20:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
21:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜
22:00	⬜	🟩	🟩	🟩	🟩	🟩	⬜

Cancel Save

3.4.4.4 USER IS A VOLUNTEER



97% 15:49

Volunteer Profile

I am a Volunteer

Available

Languages
You are available for 7 languages

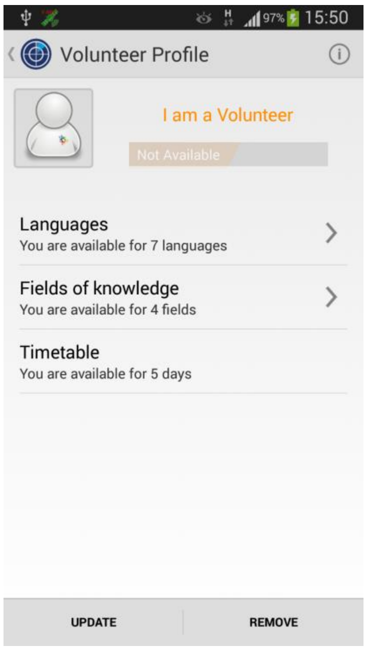
Fields of knowledge
You are available for 4 fields

Timetable
You are available for 5 days

UPDATE REMOVE

- The Volunteer can declare him/herself as temporarily unavailable by tapping on the "ON" switch so turning the switch "OFF"
- The volunteer can terminate to be a volunteer by tapping on "Remove" button

3.4.4.5 TEMPORARILY MODIFY AVAILABILITY



97% 15:50

Volunteer Profile

I am a Volunteer

Not Available

Languages
You are available for 7 languages

Fields of knowledge
You are available for 4 fields

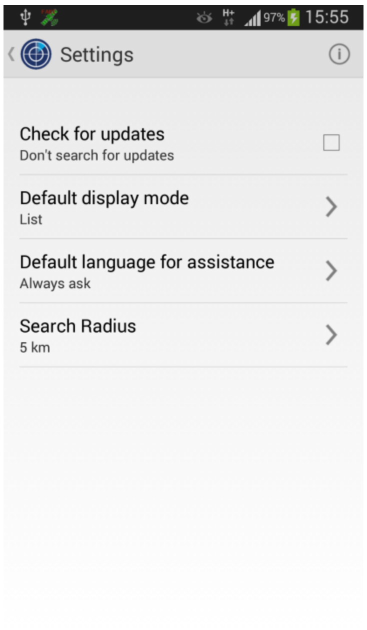
Timetable
You are available for 5 days

UPDATE REMOVE

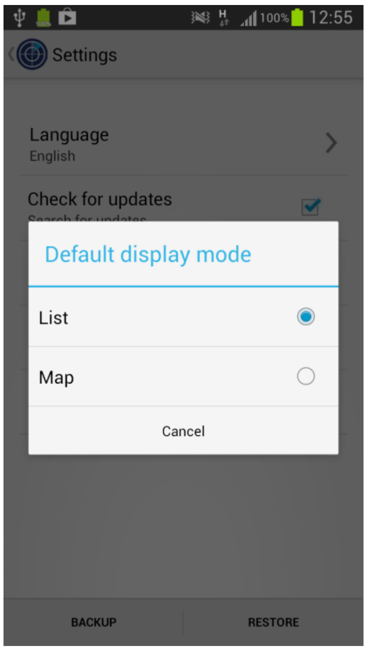
The Volunteer can declare him/herself as available by tapping on OFF switch so turning the switch "ON"

- The volunteer can terminate to be a volunteer by tapping on "Remove" button

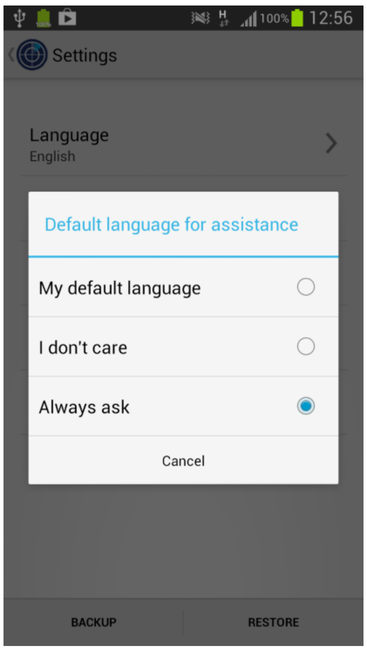
3.4.5 Settings

 Settings Check for updates Don't search for updates Default display mode List Default language for assistance Always ask Search Radius 5 km	• The user can set some application parameters

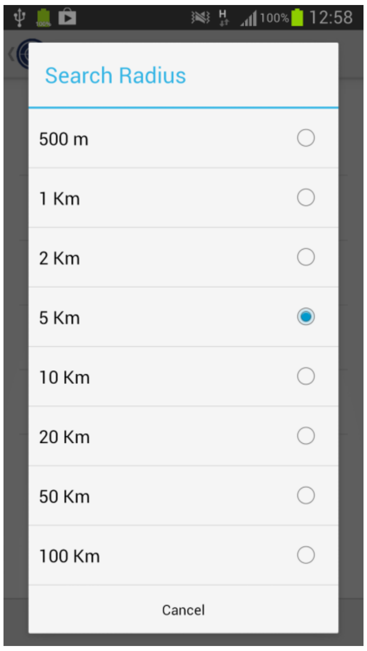
3.4.5.1 Default volunteer presentation mode

 Settings Language English Check for updates Default display mode List Map Cancel BACKUP RESTORE	The user selects volunteers default display mode (map/list)

3.4.5.2 VOLUNTEER LANGUAGE

 12:56 Settings Language English Default language for assistance My default language I don't care [✓ Always ask] Cancel BACKUP RESTORE	The user selects the language of the volunteers he/she will look for asking for assistance

3.4.5.3 SEARCH AREA

 Search Radius 500 m 1 Km 2 Km [✓ 5 Km] 10 Km 20 Km 50 Km 100 Km Cancel	The user selects the area range around his/her current position to search for volunteers

3.5 RESTFUL API

This chapter describes the APIs that will be used for communication between Help Radar application on the Android device and Help Radar application on server side.

More details on Web Services API can be found in [2].

Name	Description	Input	Output
Service Subscription	Subscribes the user to Help Radar service	Unique_ID + GCM key + validation code	Help Radar local userID
Service Unsubscription	Removes the user from Help Radar service	userID	-
Add Location Data	Sends user/volunteer location data to Help Radar server	userID + Lat + Long	-
Search volunteers	Searches volunteers located nearby the user	userID + location data + volunteer knowlegdes list + volunteer language + search area range	Volunteers list
Get volunteer details	Retrieves details about selected volunteer	userID + validation code	Volunteers Settings
Get volunteer assistance list	Provides a list of assistances performed by a volunteer	userID	Performed assistances list
New Assistance request	Creates a new Assistance request	userID + volunteerID + knowledge list	-
Signup as volunteer	Registers user as volunteer	userID + knowledge list + languages list + availability daytime	-
Update volunteer parameters	Updates volunteer parameters	userID + knowledge list + languages list + availability daytime	-
Update availability status	Updates volunteer visibility status	userID + volunteer status	-
Delete volunteer	Removes a volunteer from	userID	-

	Help Radar service		
Get assistance list	Provides a list of assistances performed for a user	userID	Received assistances list
Vote assistance	Updates user’s rate for volunteer assistance	assistanceID + rate	-
Backup User settings	Saves user settings in Help Radar server database	userID + user settings	-
Restore User settings	Restores user settings from Help Radar server database	userID	User Settings
Check for Missing Vote	Checks if user received any assistances which he/she hasn’t voted yet	userID	Boolean
Send Message	Send a message to user or volunteer	senderID + receiverID + GCM key	-
Get Chat	Retrieves all the messages exchanged during an assistance phase		Messages list
Get Open Chat	Retrieves all the message exchanged during an open assistance phase		Messages list

4. SERVER SIDE

4.1 WEB SERVER LAYER

The following paragraphs describe the Help Radar Web Server layer software components and functionalities. Starting from a description of the Web Server architecture, a complete description of their functionalities will be done.

4.1.1 ARCHITECTURE

The Web Server acts as a mediation layer between Help Radar Android application (user) and the Business layer (see Fig 1 – Overall Architecture).

The purpose of this layer is to listen for incoming requests from RestFul Web Service interface, wake up the related business components via RMI interface, and return to Android device the requested data. There is no business logic on it, and, at the time of writing, there is no Web application with administration functionalities.

The Web Server is built on two main infrastructure components:

- the JERSEY framework to manage the Web Service interface, acting as a Restful Web Service server
- the JAVA RMI framework to manage the business interface, acting as a RMI client

The Figure 4 in the following depicts the Web Server architecture.

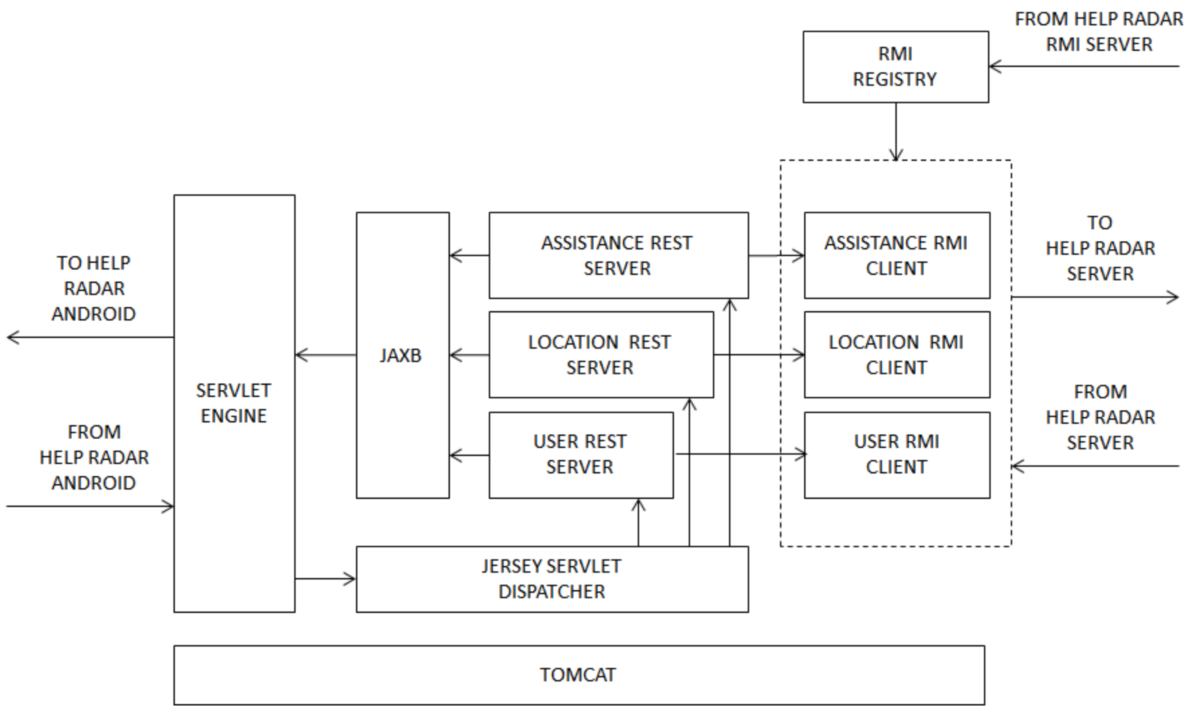


Figure 4: Web Server architecture

4.1.2 RESTFUL FRAMEWORK

4.1.2.1 JERSEY SERVLET DISPATCHER

Java defines REST support via the Java Specification Request 311 (JSR). This specification is called JAX-RS (The Java API for RestFul Web Services). JAX-RS uses annotations to define the REST relevance of Java classes.

As already mentioned in the previous paragraphs, the RestFul interface is built on the Jersey framework*:* Jersey is the reference implementation for the Java Specification Request 311. Jersey contains basically a REST server and a REST client. Although the core client can be used to communicate with the server, it is not compliant with Android Operating System, so in the Help Radar client another RestFul implementation has been used.

On the server side, the Jersey servlet dispatcher component scans predefined classes to identify RestFul resources. This component analyzes the incoming HTTP request and selects the correct class and method to respond to this request. This selection is based on JAX-RS annotations in the class and methods.

4.1.2.2 APPLICATION REST SERVER

In the Help Radar service, the class activated by the servlet dispatcher implements the following functional components:

- <u>the User REST server</u>: this component manages Help Radar users and volunteers settings and provides methods for sending messages between users and volunteers
- <u>the Assistance REST server</u>: this component manages assistance history data e messages
- <u>the Location REST server</u>: this component manages location data e users/volunteers position

All response data managed by these components are returned to Android device using the JSON syntax. JAX-RS supports the creation of JSON via the Java Architecture for XML Binding (JAXB). JAXB is able to automatically create the correct JSON structures if native JAVA classes are annotated with JAX-RS annotation syntax.

4.1.3 RMI CLIENT

This component manages interactions between the Web server layer and the Business layer via RMI protocol, acting as a bridge from the RestFul server and the Business components.

The RMI client implements the interfaces described in the following paragraph.

4.1.3.1 RMI API

This chapter describes the APIs that will be used in communication between Web Server and Business layer.

Three API interfaces have been defined:

1. <u>UserServiceInterface</u>: this interface implements methods for managing Help Radar users and volunteers(e.g. subscription, sign-up, settings,..), and for sending messages
2. <u>AssistanceServiceInterface</u>: this interface implements methods for managing assistance messages exchange
3. <u>LocationServiceInterface</u>: this interface implements methods for managing location data and volunteer searching.

4.1.3.1.1 UserServiceInterface

Name	Description	Input	Output
subscribe	Subscribes user to Help Radar service	Unique_ID + GCM key + validation code	Help Radar local userID
unsubscribe	Removes user from Help Radar service	Unique_ID	-
searchVolunteer	Searches volunteers located nearby user	userID + location data + volunteer knowlegdes list + volunteer language + search area range	Volunteer list
getVolunteerDetails	Retrieves details about selected volunteer	userID + validation code	Volunteers Settings
signupVolunteer	Registers user as volunteer	userID + knowlegdes list + languages list + availability daytime	-
updateVolunteerParams	Updates volunteer parameters	userID + knowlegdes list + languages list + availability daytime	-
updateVolunteerStatus	Updates volunteer visibility status	userID + volunteer status	-
deleteVolunteer	Removes a volunteer from Help Radar service	userID	-
saveUserSettings	Saves user settings in Help Radar server database	userID + user settings	-
getUserSettings	Restores user settings from Help Radar server database	userID	User settings
sendMessage	Send a message to user or volunteer	senderID + receiverID + message	-

4.1.3.1.2 AssistanceServiceInterface

Name	Description	Input	Output
newAssistance	Creates a new Assistance request	userID + volunteerID + knowlegdes list	-
getVolunteerAssistanceList	Provides a list of assistances performed for a user	userID	Performed assistances list
voteAssistance	Updates user's rate for volunteer assistance	assistanceID + rate	-
getUserAssistanceList	Provides a list of assistances performed for a user	userID	Received assistances list
getChat	Retrieves all the messages exchanged during an assistance phase	assistanceID	Messages list
getOpenChat	Retrieves all the messages exchanged during an open assistance phase between a particular couple of users	userID + volunteerID	Messages list
getMissingVotes	Checks if user received some assistances and he hasn't voted them yet	userID	boolean

4.1.3.1.3 LocationServiceInterface

Name	Description	Input	Output
addLocationData	Sends user/volunteer location data to Help Radar server	userID + Lat + Long	-

4.1.4 API MAPPING

This chapter describes the functional mapping between the services defined in RestFul API and the services defined in RMI API.

RestFul API	RMI API
Service Subscription	subscribe
Service Unsubscription	unsubscribe
Add Location Data	addLocalizationData
Search volunteers	searchVolunteer
Get volunteer details	getVolunteerDetails
Get volunteer assistance list	getVolunteerAssistanceList

New Assistance request	newAssistance
Signup as volunteer	signupVolunteer
Update volunteer parameters	updateVolunteerParams
Update availability status	updateVolunteerStatus
Delete volunteer	deleteVolunteer
Get assistance list	getUserAssistanceList
Vote assistance	voteAssistance
Backup User settings	saveUserSettings
Restore User settings	getUserSettings
Check for Missing Vote	getMissingVotes
Send message	sendMessage

4.2 BUSINESS LAYER

The following paragraphs describe the Help Radar Business layer software components and functionality. Starting from a description of the Business layer architecture, a complete description of its functionality will be done.

4.2.1 ARCHITECTURE

The Business layer acts as an application server. The purpose of this layer is to listen for incoming requests from RMI interface, wake up the related local business components, and return to Web Server layer the requested data. The whole Help Radar application logic is performed by this layer. Help Radar applications data and user profiles data are also managed by this layer.

The Business layer is organized in several local infrastructure and application components:

- the JAVA RMI framework, to manage the business interface, acting as a RMI server
- the application components, to manage the core Help Radar functional points
- the GCM local server, to forward users and volunteers messages to Google Cloud infrastructure
- the data layer, to manage application and user data

And one external service:

- the Google Cloud Messaging server, to deliver Help Radar messages to final destination on Android devices

Figure 5 describes the Business layer architecture.

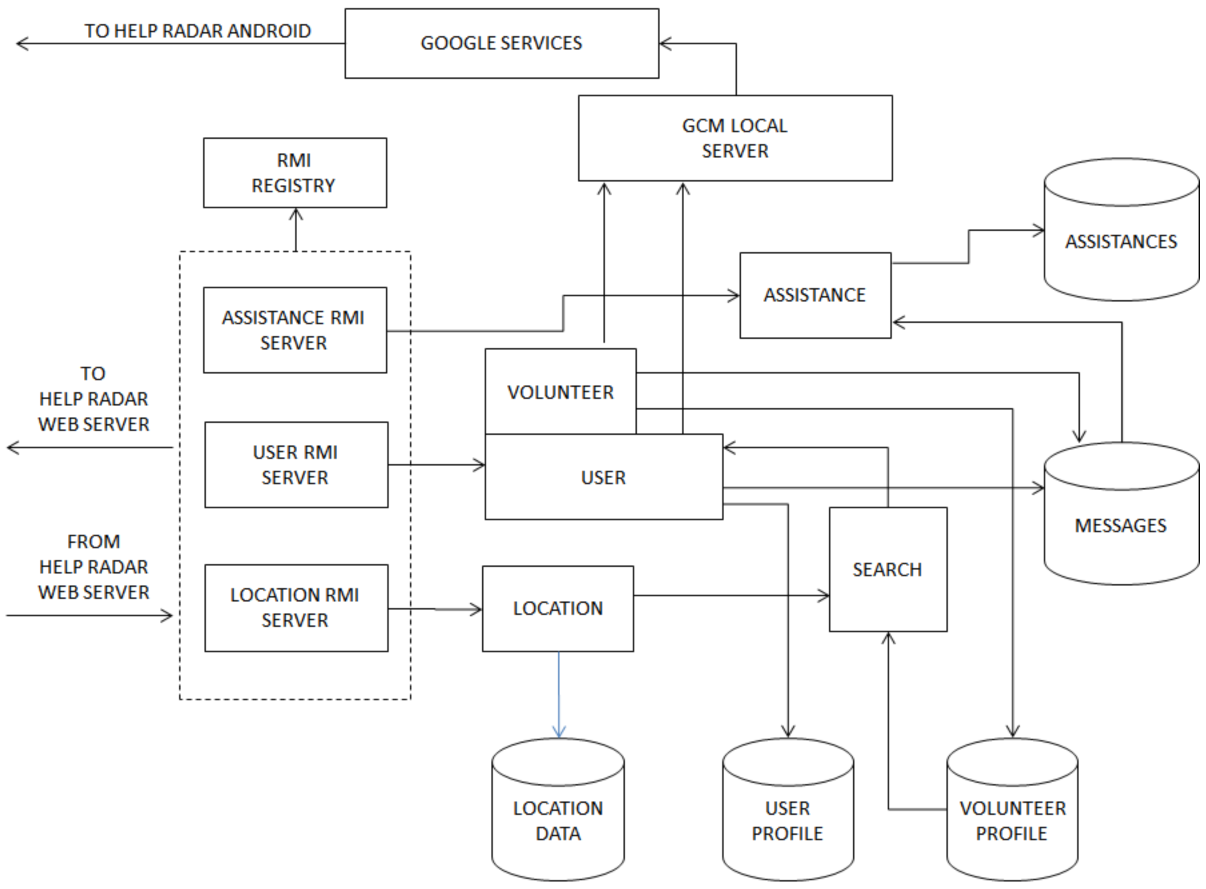


Figure 5: Business layer architecture

4.2.2 RMI SERVER

This component manages interactions between the Web server layer and the Business layer via RMI protocol, acting as a bridge from the RestFul server and the Business components.

The RMI client implements the interfaces described in the paragraph 4.1.3.1.

4.2.3 APPLICATION COMPONENTS

This paragraph describes the Business layer components and services, as depicted in Figure 5.

Name	Description
User	This component is responsible for the user subscription on Help Radar local server, for managing the user settings and message exchange during a chat session. It's activated by User RMI server component. It interacts with Search component and GCM component.
Volunteer	This component is responsible for the user registration as volunteer, for managing the volunteer settings and message

	exchange during a chat session. It's activated by User RMI server component. It interacts with GCM component.
Location	This component manages location data and computes user-to-volunteer distance. It's activated by Location RMI server and interacts with Search component
Search	This component performs volunteer search. It's activated by User component and interacts with Location component
Assistance	This component manages assistance records, volunteers rate and message exchange history. It's activated by Assistance RMI server.
GCM local server	This component forwards user and volunteer messages to Google Cloud Messaging server. It's activated by User and Volunteer components.

4.2.4 SEARCH ALGORITHM

The volunteer "Search" algorithm is mainly based on volunteer location data, that are recorded periodically (30 sec frequency) on Help Radar database, and on some user filtering criteria.

The algorithm is built on a three-steps wizard:
1. All users registered as volunteers with the required knowledge and language skills are selected
2. For each volunteer found in step1, the algorithm checks if he/she is available at the time of the search, depending on the Volunteer Availability setting (see par. 3.4.4.3).
3. For each available volunteer found in step 2, the algorithm checks if he/she is nearby the user, that is the distance between the user and the volunteer is less or equal to the value defined in the user request. The distance computation algorithm is explained in the next paragraph.

4.2.5 DISTANCE COMPUTATION ALGORITHM

The algorithm used by Help Radar to compute the user-to-volunteer distance is based on the great-circle (or orthodromic) distance formulas. The great-circle distance is the shortest distance between any two points on the surface of a sphere measured along a path on the surface of the sphere.

Because the Earth is nearly spherical, equations for great-circle distance can be used to roughly calculate the shortest distance between points on the surface of the Earth.

Let $\phi_s, \lambda_s;\ \phi_f, \lambda_f$ be the geographical latitude and longitude of two points respectively, and $\Delta\phi, \Delta\lambda$ their absolute differences; then $\Delta\widehat{\sigma}$, the central angle between them, is given by the spherical law of cosines:

$$\Delta\widehat{\sigma} = \arccos\big(\sin\phi_s \sin\phi_f + \cos\phi_s \cos\phi_f \cos\Delta\lambda\big).$$

The distance *d* for a sphere of radius *r* and $\Delta\widehat{\sigma}$ given in

$$d = r\,\Delta\widehat{\sigma}.$$

This arccosine formula above can have large rounding errors if the distance is small (less than 1 km). A more complicated formula that is accurate for all distances is the following special case (a sphere, which is an ellipsoid with equal major and minor axes) of the Vincenty formula ([4]) (which more generally is a method to compute distances on ellipsoids):

$$\Delta\widehat{\sigma} = \arctan\left(\frac{\sqrt{\left(\cos\phi_f \sin\Delta\lambda\right)^2 + \left(\cos\phi_s \sin\phi_f - \sin\phi_s \cos\phi_f \cos\Delta\lambda\right)^2}}{\sin\phi_s \sin\phi_f + \cos\phi_s \cos\phi_f \cos\Delta\lambda}\right.$$

The distance *d* for a sphere of radius *r* and $\Delta\widehat{\sigma}$ given is the same as above:

$$d = r\,\Delta\widehat{\sigma}.$$

The Vincenty formula variation is currently used in Help Radar algorithm. However, when this formula is computed using some programming languages (e.g. Java), the best choice is to use the atan2()[1] function rather than the ordinary arctangent function (atan()[2]), in order to simplify handling of the case where the denominator is zero, and to compute $\Delta\widehat{\sigma}$ unambiguously in all quadrants.

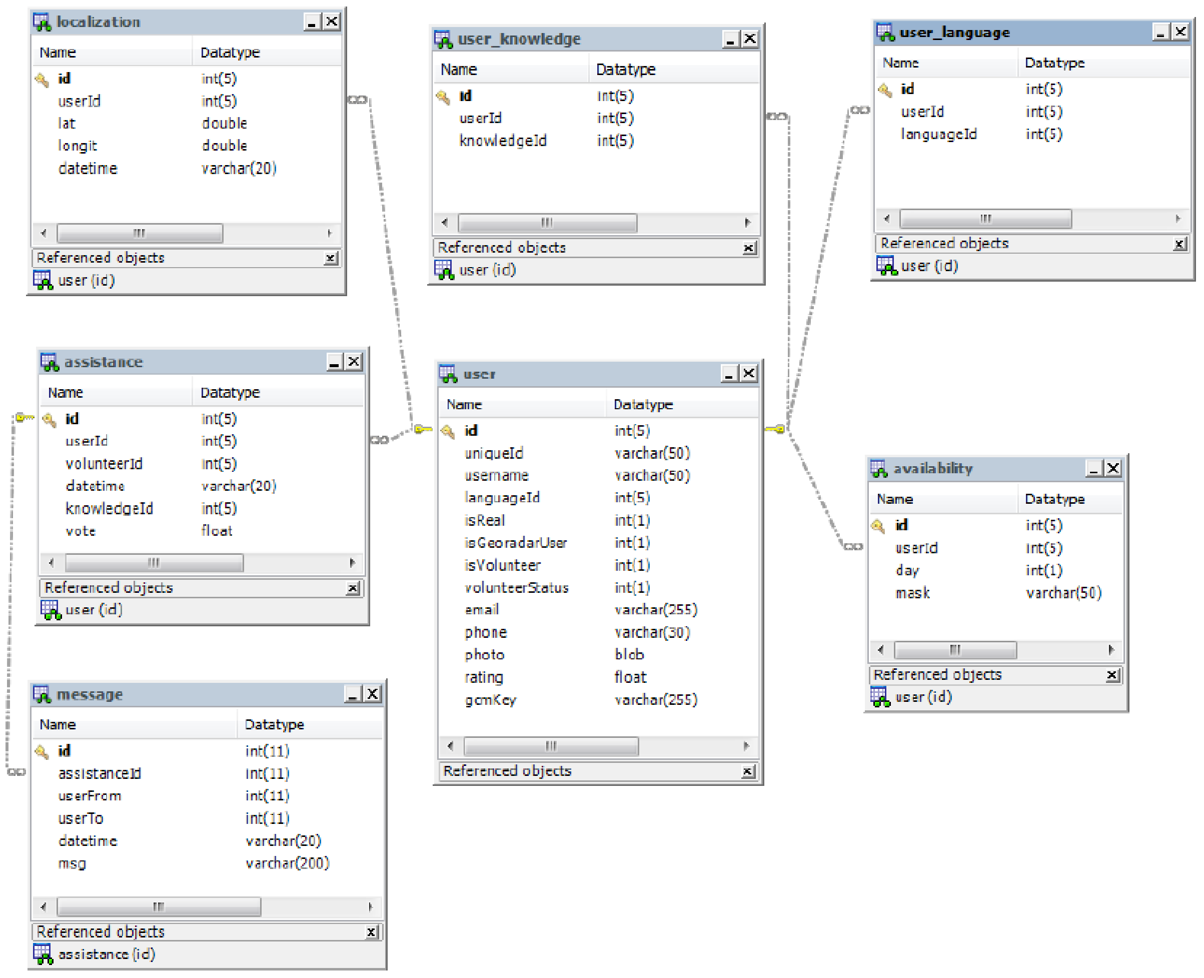
Another aspect to keep in mind when evaluating rounding errors in geodesics distance, is about the Earth radius. Considering that Earth has a non-spherical body, its radius is different if we consider equatorial radius rather than polar radius. Then, in computing distance, normally a mean radius values is used. This value is 6372,8 km.

4.2.6 DATA LAYER

Data layer is built on a Mysql database engine. This paragraph describes the structure of the Help Radar database and the core tables, as depicted in Figure 6 diagram.

Services tables, e.g. lookup tables, are not described.

[1] atan2: Returns the angle *theta* from the conversion of rectangular coordinates (x, y) to polar coordinates (r, *theta*).
[2] atan: Returns the arc tangent of a value; the returned angle is in the range -*pi*/2 through *pi*/2.



localization

Name	Datatype
id	int(5)
userId	int(5)
lat	double
longit	double
datetime	varchar(20)

Referenced objects: user (id)

user_knowledge

Name	Datatype
id	int(5)
userId	int(5)
knowledgeId	int(5)

Referenced objects: user (id)

user_language

Name	Datatype
id	int(5)
userId	int(5)
languageId	int(5)

Referenced objects: user (id)

assistance

Name	Datatype
id	int(5)
userId	int(5)
volunteerId	int(5)
datetime	varchar(20)
knowledgeId	int(5)
vote	float

Referenced objects: user (id)

user

Name	Datatype
id	int(5)
uniqueId	varchar(50)
username	varchar(50)
languageId	int(5)
isReal	int(1)
isGeoradarUser	int(1)
isVolunteer	int(1)
volunteerStatus	int(1)
email	varchar(255)
phone	varchar(30)
photo	blob
rating	float
gcmKey	varchar(255)

Referenced objects

availability

Name	Datatype
id	int(5)
userId	int(5)
day	int(1)
mask	varchar(50)

Referenced objects: user (id)

message

Name	Datatype
id	int(11)
assistanceId	int(11)
userFrom	int(11)
userTo	int(11)
datetime	varchar(20)
msg	varchar(200)

Referenced objects: assistance (id)

Figure 6: Database core diagram

<u>User table:</u>

Column name	Description
id	primary key
uniqueID	MASELTOV unique user identifier
username	user nickname
languageId	id of user language
isGeoradarUser	flag indicating the user subscription at Help Radar service
isVolunteer	flag indicating the user registration as volunteer
volunteerStatus	volunteer status (hide, active)
email	volunteer email
phone	volunteer phone number
rating	volunteer rating
gcmKey	Google Cloud Messaging identifier

<u>Localization table:</u>

Column name	Description
id	primary key
userId	user identifier
lat	user latitude
longit	user longitude
datetime	date and time of receiving the user position data on Help Radar server

Assistance table:

Column name	Description
id	primary key
userId	user identifier
volunteerId	volunteer identifier
datetime	date and time of opening of the new assistance request
knowledgeId	knowledge for which the user needs help
vote	assistance rate

Message table:

Column name	Description
id	primary key
assistanceId	assistance unique identifier
userFrom	identifier of sender user
userTo	identifier of receiving user
datetime	date and time of sending message
Msg	the message

Availability table:

Column name	Description
id	primary key
userId	user identifier
day	day of week
mask	24 hours availability mask for the day

5. INSTALLATION REQUIREMENTS

5.1 CLIENT

As mentioned in previous chapter, Help Radar apklib has to be integrated into MASELTOV application. So, Help Radar installation must meet the whole application requirements.

On the other hand, some specific requirements must be kept in mind to correctly install and run Help Radar client component.

The following paragraphs describe hardware and software requirements for Help Radar component.

The Help Radar APKLIB is available on the following Telecom Italia Maven Repository:

http://sesamo.tilab.com/snapshots

5.1.1 HARDWARE REQUIREMENTS

CPU core	>=2
Display size	4’’ or wider
Internet connectivity	HSDPA
Sensors	GPS, WiFi

5.1.2 SOFTWARE REQUIREMENTS

Operating System	Android 4.1.1.4 or higher
JAVA Platform	J2ME

5.2 SERVER

5.2.1 HARDWARE REQUIREMENTS

CPU core	>=2
RAM	4 GB
HD	50 GB

If the server runs behind a firewall, the following ports must be opened:

- HTTP 80 (input)
- SSL 443 (output)

5.2.2 SOFTWARE REQUIREMENTS

Operating System	Windows 7 Enterprise Edition - Service Pack 1 - 64-bit
JAVA Platform	JDK 1.6

Servlet Container	Tomcat 7
Database engine	MySQL 5.5

5.2.3 INSTALLATION

The required components to install Help Radar server (both Web layer and Business layer) are available on the following Telecom Italia SVN Repository:
http://polaris.tilab.com/svn/maseltov-public project "GeoRadarServerDist".

The following table describes the components.

Item	Description
GeoRadarServer_lib	this folder contains the libraries used by Business layer
script	this folder contains the installation and execution batch
GeoRadarServer.jar	it contains the Business layer classes and resources
GeoRadarWeb.war	it contains the Web layer classes and resources
GeoRadarWS.war	it contains the Web service interface classes and resources

5.2.3.1 INSTALLATION PROCESS

1. Download components from TI SVN Repository and copy them in a folder (eg. C:\GeoRadar\deploy) of the target machine
2. Configure batch parameters as described in par. 5.2.3.2
3. Execute installWeb.bat batch to install Web layer under Tomcat
4. Execute startRegistry.bat batch to start the RMI Registry
5. Execute startServer.bat batch to start the Business layer
6. Execute startTomcat.bat bath to start Tomcat

5.2.3.2 BATCH CONFIGURATION

All the scripts deployed on the target machine must be configurated before running.

5.2.3.2.1 installWeb.bat

This batch installs the Help Radar Web application and the Help Radar Web Service application under Tomcat. The following parameters must be configured:

Parameter	Description
JAVA_HOME	Path to Java JDK
TOMCAT_HOME	Path to Tomcat installation
GEORADAR_HOME	Path to Help Radar deploy folder (eg. C:\GeoRadar\deploy)

5.2.3.2.2 cleanWeb.bat

This batch removes old versions of Help Radar Web application and Help Radar Web Service application from Tomcat, if any. The following parameters must be configured:

Parameter	Description
JAVA_HOME	Path to Java JDK
TOMCAT_HOME	Path to Tomcat installation

5.2.3.2.3 startRegistry.bat

This batch starts the RMI Registry on the target machine. The following parameter must be configured:

Parameter	Description
JAVA_HOME	Path to Java JDK

5.2.3.2.4 startServer.bat

This batch starts the Help Radar Business layer on the target machine. The following parameters must be configured:

Parameter	Description
JAVA_HOME	Path to Java JDK
GEORADAR_HOME	Path to Help Radar deploy folder (eg. C:\GeoRadar\deploy)

5.2.3.2.5 startTomcat.bat

This batch starts Tomcat on the target machine. The following parameters must be configured:

Parameter	Description
JAVA_HOME	Path to Java JDK
TOMCAT_HOME	Path to Tomcat installation

5.2.3.2.6 stopTomcat.bat

This batch stops Tomcat on the target machine. The following parameters must be configured:

Parameter	Description
JAVA_HOME	Path to Java JDK
TOMCAT_HOME	Path to Tomcat installation

6. SUMMARY AND OUTLOOK

The Help Radar software platform is made up by a client service, running as a part of MASELTOV Android app, and a back-office component running on a separate server.

Each user has to subscribe to the Help Radar service. After subscription, Knowledge Profile of volunteers can be searched. As search result, a list will be shown to the user with the volunteers found. User can look at volunteer's detailed info (language, proximity, knowledge, rating) and get in contact with the selected one

User can sign up himself as a volunteer, defining his special knowledge/profession (law, mechanic, …), and setting some other parameters for better profiling the service: the accuracy of the position provided and the daytime availability. Availability can be changed whenever needed by the user.

User can view his/her history of received assistance by volunteers. He can vote the satisfaction level about the assistance received. For each aid, involved volunteer and volunteer rating are shown.

Based on user request for assistance, the server component will localize the current user position as well as volunteers current position for those volunteers that meet user requirement in terms of knowledge. To know which volunteers meet user assistance needs, the Help Radar Platform will get user profiles from User Profiling component. If some volunteers are found nearby user position, a brief list of them will sent back to user. Also a detailed user information could be sent to the user.

If a user wants to sign up himself/herself as volunteer, he/she must have a Validation Code provided by NGOs. The Help Radar Platform will collect all specific data (knowledge, language, daytime availability,..), saving them into local database and updating the MASELTOV user profile. Other information can be saved by the platform (user status, current local position).

Assistance request history is also managed by the platform: for each request, a new assistance record will be added to local database. Based on user request, these records can be sent to user and visualized on the smartphone.

The final version of Help Radar service will be evaluated during the final filed trials (D9.4).

7. REFERENCES

[1] Patrick Luley (JR), Stefan Ladstätter (JR), Anna Weber (JR), Mirjana Artukovic (FLU), Ian Dunwell (COV), Lukas Neumann (CTU), Mark Gaved (OU), Nicoletta Bersia (TI), Charlie Pearson (PP), Sofoklis Efremidis(AIT). MASELTOV Deliverable Report D3.3.3 “Iterative System Integration”

[2] Patrick Luley (JR), Lucas Paletta (JR), Mirjana Artukovic (FLU), Bilal Abunaim (FLU), Ian Dunwell (COV), Lukas Neumann (CTU), Mark Gaved (OU), Percannella Gianraffaele (TI), Massimo Cappello (TI), Sofoklis Efremidis(AIT). MASELTOV Deliverable Report D3.2.2 “System Specification”

[3] Patrick Luley (JR), Lucas Paletta (JR), Mirjana Artukovic (FLU), Bilal Abunaim (FLU), Ian Dunwell (COV), Lukas Neumann (CTU), Mark Gaved (OU), Percannella Gianraffaele (TI), Massimo Cappello (TI), Sofoklis Efremidis(AIT). MASELTOV Deliverable Report D3.1.2 “System Architecture”

[4] Thaddeus Vincenty. “Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations”

[5] http://developer.android.com/google/gcm/index.html

[6] Massimo Cappello (TI). MASELTOV Deliverable Report D8.2.1 “Geosocial Mobility and Communication Model”

[7] Lucas Paletta (JR), Patrick Luley (JR), Martina Uray (JR), Jan Bobeth, Susanne Schmehl (CURE), Lazaros Polymenakos (AIT), Adela Ros (UOC), Mark Gaved (OU), Ian Dunwell (COV), Jiri Matas (CTU), Walter Scheitz (FHJ), Graziella Spinelli, Nicoletta Bersia (TI), Mirjana Artukovic (FLU), George Pearson (PP), Sara Wickert (MRC), Marianne Hammani-Birnstingl (DAN), Samuel Ricardo Ruiz (FUN). MASELTOV Deliverable Report D1.1.4 Periodic Annual Progress Report and Management Summary”

[8] Patrick Luley (JR), Stefan Ladstätter (JR), Anna Weber (JR), Mirjana Artukovic (FLU), Ian Dunwell (COV), Lukas Neumann (CTU), Mark Gaved (OU), Nicoletta Bersia (TI), Charlie Pearson (PP), Sofoklis Efremidis(AIT). MASELTOV Deliverable Report D6.1.2 “Mobile Assistant Service”

[9] Lucas Paletta (JR), Patrick Luley (JR), Martina Uray (JR), Jan Bobeth, Susanne Schmehl (CURE), Lazaros Polymenakos (AIT), Adela Ros (UOC), Mark Gaved (OU), Ian Dunwell (COV), Jiri Matas (CTU), Walter Scheitz (FHJ), Graziella Spinelli(TI), Nicoletta Bersia (TI), Mirjana Artukovic (FLU), George Pearson (PP), Sara Wickert (MRC), Marianne Hammani-Birnstingl (DAN), Samuel Ricardo Ruiz (FUN). MASELTOV Deliverable Report D1.1.4 “Periodic Annual Progress Report and Management Summary”