



Project acronym: **Go-myLife**

Project full title: **Going on line: my social Life**

AAL Joint Programme



Call for Proposals AAL-2009-2-089

## **D4.1 Platform Prototype 1**

Author: Idoia Olalde (Andago)

Version: 1.0

Date: 27/02/2012

<b>Deliverable Number:</b>	D4.1
<b>Contractual Date of Delivery:</b>	31/08/2011
<b>Actual Date of Delivery:</b>	27/02/2012
<b>Title of Deliverable:</b>	Platform Prototype 1
<b>Dissemination Level:</b>	Public
<b>WP contributing to the Deliverable:</b>	WP4
<b>Author(s):</b>	Idoia Olalde (Andago)
<b>Participant(s):</b>	Ioannis Kouris (ICCS) / Fabio Tumiatti (Atos) / Andrew McDonough (Atos) / Sergio García (Atos) / Mikel Serrano (Andago) / Roberto Calvo (URJC)

History			
Version	Date	Author	Comments
0.0	15/12/2011	Idoia Olalde (Andago)	Draft version
0.1	23/12/2011	Ioannis Kouris (ICCS) Fabio Tumiatti (Atos) Andrew McDonough (Atos)	Initial contributions
0.2	01/02/2012	Ioannis Kouris (ICCS) Sergio García (Atos) Roberto Calvo (URJC) Mikel Serrano (Andago)	Contributions
0.3	21/02/2012	Idoia Olalde (Andago)	First complete version
1.0	27/02/2012	Idoia Olalde (Andago)	Final version

Approval and Sign-off		
Date	Name	Sign-off
27/02/2012	Michael Mulquin	Approved

**Abstract**

This report analyses the potential usage of the Go-myLife platform and its benefits both for end-users and stakeholders. It also describes the Go-myLife prototype implemented for the first trials, the functionalities offered by the platform and the user interface. It also describes the set of interfaces and tools to facilitate the integration of services and content from external sources and the creation of communities.

**Keywords**

Go-myLife architecture, Go-myLife platform, API, Go-myLife prototype.

# Table of Contents

1	Introduction.....	8
1.1	Summary.....	8
1.2	Role of this deliverable .....	8
2	Go-myLife architecture design .....	9
2.1	Go-myLife services and components for end-users .....	11
2.1.1	Local Life.....	11
2.1.2	Around Me .....	11
2.2	Go-myLife services and components for stakeholders.....	12
2.2.1	Services Integration Interface .....	12
2.2.2	Communities Integration Interface .....	12
2.2.3	Data Analysis Interface.....	12
2.2.4	3 <sup>rd</sup> Party Customized Interfaces .....	13
3	Implemented functionalities in the first prototype.....	13
4	Prototype design.....	72
5	APIs, SDKs and tools .....	85
5.1.1	User interface API.....	85
5.1.2	Services integration API .....	89
5.1.3	Communities Integration API.....	92
5.1.4	External Social Networks API.....	94
5.1.5	Data Analysis API.....	95
6	Conclusion .....	96
7	Annex 1 Creative Commons images' references.....	97

## Tables of Figures

Figure 1 Go-myLife Platform Architecture .....	10
Figure 2: Login form (Desktop).....	14
Figure 3: Login form (Mobile) .....	15
Figure 4: Logout icon in the upright section of the main screen (Desktop) .....	16
Figure 5: Logout icon in the upright section of the main screen (Mobile) .....	17
Figure 6: Registration form (Desktop).....	18
Figure 7: My basic profile info (Desktop) .....	19
Figure 8: My basic profile info (Mobile).....	20
Figure 9: Profile picture manager (Desktop) .....	21
Figure 10: Profile picture manager (Mobile).....	22
Figure 11: User location (Mobile) .....	23
Figure 12: General information form (Desktop).....	24
Figure 13: Contact info form (Desktop) .....	25
Figure 14: Change user password (Desktop).....	26
Figure 15: User basic info (Desktop).....	27
Figure 16: User basic info (Mobile).....	28
Figure 17: My network group selection (Desktop).....	29
Figure 18: My network group selection (Mobile).....	30
Figure 19: User group members (Desktop) .....	31
Figure 20: User group members (Mobile) .....	32
Figure 21: User friends (Desktop) .....	33
Figure 22: User friends (Mobile) .....	34
Figure 23: Friendship invitations (Desktop).....	35
Figure 24: Friendship invitations (Mobile).....	36
Figure 25: The form for adding groups (Desktop).....	37
Figure 26: The form to add members into a group (Desktop) .....	38
Figure 27: Delete group list (Desktop) .....	39
Figure 28: The News screen, writing a post (Desktop) .....	40
Figure 29: The News screen, writing a post (Mobile) .....	41
Figure 30: Posting a comment to a piece of news (Desktop) .....	42
Figure 31: Posting a comment to a piece of news (Mobile) .....	43

Figure 32: A commented piece of news (Desktop) .....	44
Figure 33: A commented piece of news (Mobile) .....	45
Figure 34: Main media screen (Desktop).....	46
Figure 35: Media gallery screen (Mobile) .....	47
Figure 36: The Upload Photo screen (Desktop) .....	48
Figure 37: The Upload Photo screen (Mobile) .....	49
Figure 38: View of a photo (Desktop) .....	50
Figure 39: View of a photo details (Mobile) .....	51
Figure 40: View of a photo comments (Mobile) .....	52
Figure 41: View Panoramio photos (Mobile) .....	53
Figure 42: List of events (Desktop) .....	54
Figure 43: List of events (Mobile) .....	55
Figure 44: Event details (Desktop) .....	56
Figure 45: Event details (Mobile).....	57
Figure 46: Users invited to the event (Mobile).....	58
Figure 47: The screen for creating events (Desktop).....	59
Figure 48: Search results (Desktop).....	60
Figure 49: Looking for a non-friend and sending request (Desktop) .....	61
Figure 50: Help in media section (Desktop) .....	62
Figure 51: Help in media section (Mobile).....	63
Figure 52: Around Me (Mobile) .....	64
Figure 53: Polish interface and language flags located in the up-right corner (Desktop) .....	65
Figure 54: The Local Life main screen (Desktop).....	66
Figure 55: The Local Life groups (Mobile).....	67
Figure 56: Culture groups and the option to join or view them (Desktop).....	68
Figure 57: Join a group (Mobile) .....	69
Figure 58: Group details (Desktop) .....	70
Figure 59: Group details (Mobile) .....	71
Figure 60: List of friends .....	73
Figure 61: Help in Polish. ....	74
Figure 62: My Network screen, showing the different groups where friends are classified. .....	75
Figure 63: My Network screen when the user has selected a group.....	76

Figure 64: Me section, showing a blue navigation bar, in concordance with its button...	77
Figure 65: Help on the Media section.....	78
Figure 66: My Network section, showing a non-scrollable list. ....	79
Figure 67: Easy profiling. ....	80
Figure 68: Mobile Login Screen .....	81
Figure 69: Mobile Home Screen .....	82
Figure 70: My Network Mobile.....	83
Figure 71: Help Screen .....	84
Figure 72: Scrollable List.....	84
Figure 73: The External Channel Aggregator.....	89
Figure 74: External Social Network API schema .....	94

# 1 Introduction

## 1.1 Summary

The aim of this deliverable, *D4.1 Platform Prototype 1*, is to document the first prototype implemented during the first cycle of the project. As part of the documentation, the Go-myLife architecture is specified as well as its relationship with end-users and potential business models. The Go-myLife platform provides a set of tools and application programming interfaces (hereinafter API) for the inclusion of services and content from external providers and for the construction of communities. This deliverable also explains the functionalities implemented in the current prototype that will be tested during the first trials with end-users and that will be the basis for the tasks of the work package (hereinafter WP) 5, where the communities and the connectors to external sources will be developed.

After studying the results and feedback of the pilots, a new set of requirements and improvements will be defined to design a revised Go-myLife architecture and develop a final version of the prototype.

## 1.2 Role of this deliverable

This deliverable *D4.1 Platform Prototype 1* is the result of the task *T4.1 Platform Prototype 1* of WP4. In this task the first Go-myLife prototype was implemented in accordance with the architecture design specified in *D3.1 Initial Platform Architecture and Design* and the prototype design explained with greater detail in *D3.2 First Prototype Design*. The output of the current task is a functioning prototype that will be tested with end-users in the pilots that will be carried out in United Kingdom and Poland as part of WP6, *Evaluation and validation through scenarios*.

This deliverable aims to summarise the Go-myLife architecture as well as the tools and API provided for the building of the local life communities that will be carried out during WP5 – *Communities construction and customization*.

The final version of the architecture, *D3.3*, will lead to the second iteration of the implementation that will be carried out in *T4.1 Platform Prototype 2*.

All technical partners have contributed to this deliverable.



## 2 Go-myLife architecture design

Go-myLife is a social networking platform based on location awareness and mobility, built around the interests and requirements of older people and integrated with other social networks.

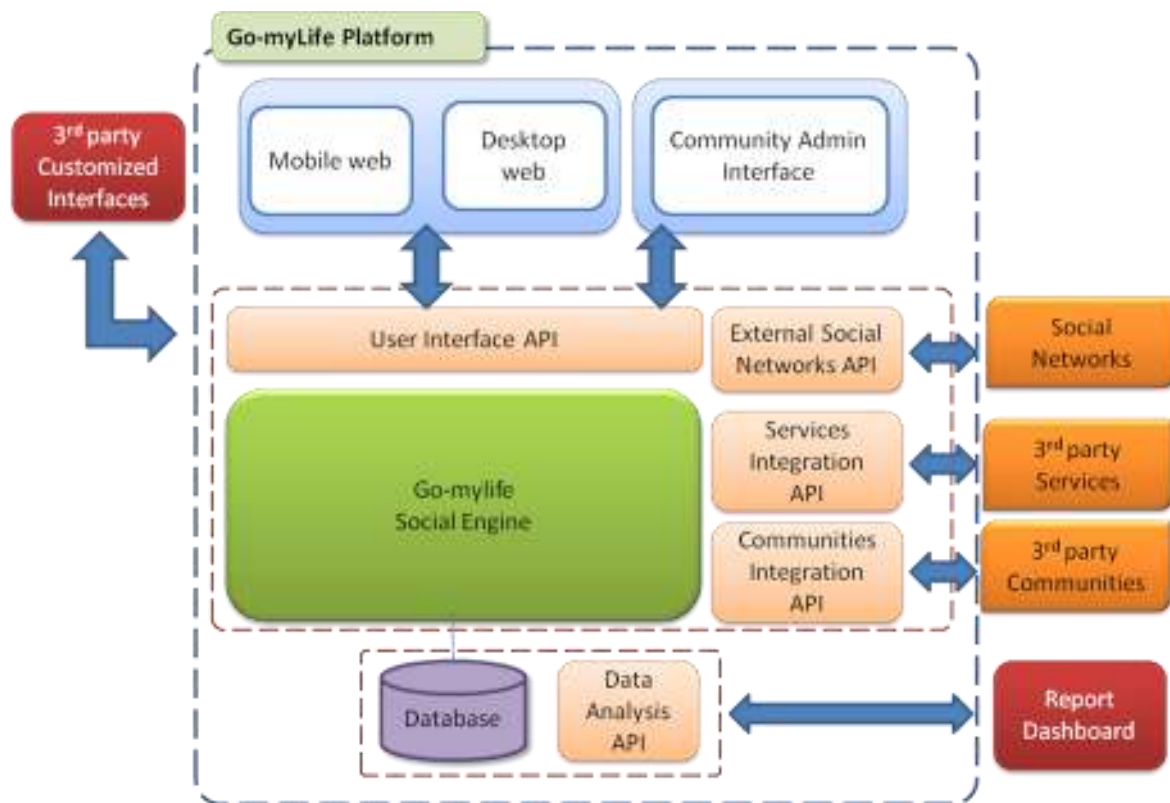
In order to manage the social content of the platform, a social engine was required. After a deep analysis of different engines that was carried out in WP2 during the technical requirement analysis, we selected LibreGeoSocial framework as the basis for the Go-myLife Social Engine. LibreGeoSocial framework is an open source framework that allows the creation of geolocated social networks, managing all the content as nodes. Each content object (user, photo, video, etc.) is represented by its latitude, longitude and even altitude.

The Go-myLife Social Network Engine exports a REST API to manage all the functionalities (User Interface API). Representational state transfer (REST) is a style of software architecture for distributed systems consisting of a client-server approach where a client makes requests to a server to obtain the data. REST interfaces guarantee the technology independence in the development of both sides. This approach provides us with the necessary flexibility for the development of any client and the opportunity for the development of consulting services to create customized client applications on demand.

One key aspect of social networks is the number of participants. Limiting access to users of just one operating system or device would limit the number of potential participants and would therefore have gone against what we considered a success criterion of the project. Taking into account the need of a social platform to be accessed ubiquitously, the consortium therefore agreed that Go-myLife's end-product would be a web application which users could interact with and access from any device; including both desktop and mobile phone. Due to the fact that a REST interface was used to manage the content in the core of the platform, there were no restrictions in the implementation of the client application.

Go-myLife platform is not a standalone social network that isolates older people in a specific social network, but it interacts with and integrates existing content and communities. It therefore provides a rich and flexible platform, allowing older people to select the content and communities they want to interact with.

Around the Go-myLife Social Engine, a set of interfaces has been built to offer an easy and standardized way to integrate external content into the platform. In that way, existing communities can be part of Go-myLife through the Communities Integration API and can create a space where users can share content and interact amongst themselves. External information providers (both commercial and non-commercial) can integrate their geolocated content into the Go-myLife platform through the Service Integration API in order to provide relevant information around the user and thus offer a value-added service. The Go-myLife platform also offers a Data Analysis API to provide statistics and information on trends in order to help 3<sup>rd</sup> party service providers analyse their services and study their competitiveness. Finally, connectors to existing online social networks can be built to facilitate bi-directional communication between the platforms.



**Figure 1 Go-myLife Platform Architecture**

Hence, the Go-myLife platform architecture, as it is illustrated in Figure 1 is composed of the following components:

- Web client application: At the user interface level, a web application, that includes a Mobile web form as well as a Desktop web, was developed using Google Web Toolkit. The Web client application provides users with web-based access to the system to interact with their peers, share content, connect to other social networks and be active members in the communities.
- Community Admin Interface: a web interface that provides administrators with web-based access to capabilities that include:
  - Managing Communities
  - Managing News
  - Managing Photos
  - Managing Events
  - Managing Location
- User Interface API: an interface to manage the content stored in the Go-myLife Social Engine. This interface is called from the UI level to create the different functionalities offered in the client application.
- Go-myLife Social Engine: The Go-myLife Social Engine is the core of the Go-myLife platform and is responsible for the management of the content, its

relationships and privacy. It is based on the open source LibreGeoSocial framework.

- External Social Networks API: an interface to connect to existing online social networks and allow bi-directional communication between the platforms. As part of the scope of Go-myLife project, different connectors have been developed for interaction with each of these networks. All of them follow a schema, to guarantee their integration without unnecessary changes at the user interface.
- Services Integration API: an interface to integrate content from external information sources. This module allows an easy integration system for sources of information; it qualifies the contents and provides clients with a good abstraction model.
- Communities Integration API: an interface for the construction of the communities and the management of their content.
- Data Analysis API: an interface to analyse the data, obtain statistics anonymously and detect trends, to be offered through a Report Dashboard to 3<sup>rd</sup> parties.
- Database: is the database software used to store all the Go-myLife related data.

## **2.1 Go-myLife services and components for end-users**

The architecture of the Go-myLife platform enables the end-users to receive personalized content, customized to their personal preferences and interests, related to the groups they participate in and the services provided around their actual location.

### **2.1.1 Local Life**

Utilizing the Communities Integration API, public bodies, local clubs and associations will provide updated content to the platform. Additionally, via the Services Integration API, the communities integrated with the platform will provide their available services to the end-users, allowing each user to receive an integrated experience through their use of the platform.

The older person using the platform will be able to review the communities that potentially meet his/her interests and that he or she might like to be part of. He or she will also be able to access available local services and interact with them in an easier and more efficient way. Using the search functionality of the platform, the older person will almost instantaneously be able to gain information regarding the activities and news of their communities, and be able to browse available services so as to interact with them in a more intuitive way.

### **2.1.2 Around Me**

The geolocation feature, whether the platform is accessed via desktop or mobile devices, provides the end-user with information regarding available services and content around his or her present location. When the user selects the option of being provided with the content and the services around him, a list of all the available communities and local

services (sorted into distinct categories) found within a predefined radius become available. In that sense, he or she can be informed about local news, be provided with a list of the clubs related to his or her needs or interests and have access to public or commercial services in the vicinity. This concept could resemble something like a car navigation system, though the Go-myLife platform provides access to content which is continuously updated and with which the user may interact in a more sensible way.

## **2.2 Go-myLife services and components for stakeholders**

This section will describe the different components of the Go-myLife architecture that can be used by external providers to provide services and content for the Go-myLife platform.

### **2.2.1 Services Integration Interface**

A Services Integration API will be provided to offer functionality to 3<sup>rd</sup> party service providers so that they may take advantage of the Go-myLife social network to offer services to Go-myLife users. Those services can be defined as consisting of two types: non-commercial and commercial. The non-commercial services are the ones that will help the user to find services that are not for commercial purposes, like the nearest hospital, pharmacies or toilets; and the commercial services are the ones for commercial purposes, like the nearest restaurants, bars or shopping malls.

The API consumes geolocated content information thus providing relevant location based information around the user. This new layer will be added to the Layer Manager of the Go-myLife Social Engine and its content will be managed as geolocated nodes.

The Go-myLife users can find those services directly by accessing the “Around me” functionality or indirectly by accessing a community that has access to those services.

### **2.2.2 Communities Integration Interface**

The community functionality is especially important to the Go-myLife project, the whole ethos of communities is employed in the Go-myLife social network with the aim of improving the social aspects of the older generation.

The Communities Integration API enables 3<sup>rd</sup> party providers to build communities and send content directly, without the need to connect via Go-myLife user interface and add data manually. The interface can also be used for advertising and marketing purposes.

### **2.2.3 Data Analysis Interface**

The Data Analysis API offers a suite of queries which may be used to provide any type of report, regarding the usage and the performance of the Go-myLife platform and additionally access directly anonymous data regarding the users. The Report Dashboard enables market analysis to be performed, as well as providing useful information on the usage of the Go-myLife platform and its various aspects. The reports generated may be predefined or customised, in order to fulfil the specific needs of the analysis. User data protection is a major concern of the Go-myLife platform, so multiple levels of security will be implemented in order to prevent inappropriate usage of the Data Analysis API.

### 2.2.4 3<sup>rd</sup> Party Customized Interfaces

As Go-myLife is open source software, if an entity wants to build a customized User Interface for their own use, using the Go-myLife Social Engine, they can do so using the User Interface API.

The User Interface API will allow both public and private entities to engage with Go-myLife users, making possible the use of any technology to build a new User Interface for accessing the Go-myLife Platform.

The added value of being able to offer customisable interfaces to third parties enhances the commercial viability of the platform, along with the opportunity to provide them with consulting and training services. High quality support tools and feedback services are also important for sustainability and future-proof investment.

## 3 Implemented functionalities in the first prototype

In previous documents, there was a description of functionalities that the final Go-myLife prototype must include. In this first prototype, there are several functionalities implemented, divided into sections and dependent on the media the user is accessing Go-myLife from (*Mobile* or *Desktop*):

**Login/Register/Logout:** Registration is mandatory for accessing Go-myLife, in order to protect the private information that the Go-myLife users share among themselves.

- Common functionalities (Mobile and Desktop)
  - Login using user name and password: when a user enters Go-myLife, a login page asking for his/her username and password is shown. Go-myLife remembers the user's session (until he or she logs out) so the next time the user enters the application, he will be directed to the main page.



Figure 2: Login form (Desktop)



**Figure 3: Login form (Mobile)**

- Reset user password service: the normal “forgotten password” link is also provided as seen in Figure 2 and Figure 3.
- Logout: If the user wishes to logout, a logout icon is provided. In the Desktop version, this icon is accessible from everywhere. In the mobile version, however, the icon is only located in the main menu.



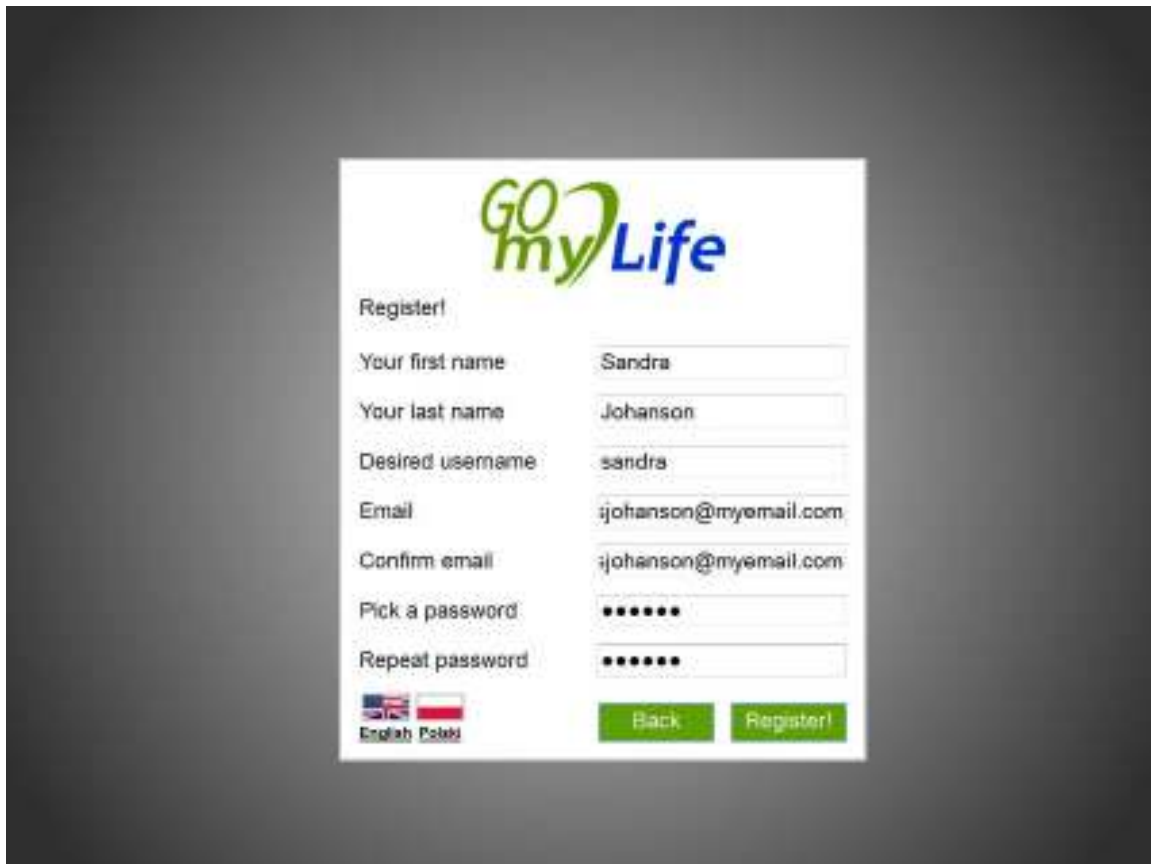
Figure 4: Logout icon in the upright section of the main screen (Desktop)





Figure 5: Logout icon in the upright section of the main screen (Mobile)

- Desktop
  - Complete registration process: if the user is not registered on the platform, a *Register* button is also provided. When it is clicked, the login form is transformed into a simple registration form, without the need for any kind of navigation actions, maintaining the simplicity we wish to achieve. Parameters needed: user name, first name, last name, email and password.



**Figure 6: Registration form (Desktop)**

**Me:** profile section where the user can view/edit his or her personal information

- Common functionalities (Mobile and Desktop)
  - View basic profile information: the user can see his or her personal information, such as email, full name and profile photo.



Figure 7: My basic profile info (Desktop) <sup>1</sup>

<sup>1</sup> All pictures shown in the screenshots have a Creative Commons license with commercial use. This deliverable describes the first Go-myLife prototype implemented during the first phase of the project and that will be tested during the pilots in UK and Poland. The Go-myLife prototype consists of the Go-myLife Social Network Engine, a set of APIs and web-based client applications to be accessed through the desktop or the mobile.

The Go-myLife Social Engine is based on the *LibreGeoSocial* framework and is the responsible for the management of the content, its relationship and its privacy. Over it, a set of APIs have been built in order to facilitate the integration of external content and the construction of communities.

The Go-myLife Social Engine offers its functionality through a REST API (User Interface API). Making use of it, we have built the web-based applications (desktop and mobile) where the users interact and share their content. Different social functionalities are offered in this first version: manage your contacts (My Network), share posts (News), share photos (Media), organize events (Events) or update your profile information (Me).

But Go-myLife takes the social network one step further and provides information about who and what is around the user (Around me), where the end-users can find local services, friends or content uploaded by others. Through the Services Integration API,

---

local companies or associations can integrate their services into Go-myLife in order to be found by the users.

Go-myLife also recognizes the relevance of existing social networks where people within the networks of the end-users may already have an account. Go-myLife doesn't want to create an isolated and exclusive social network for older people but it wants to offer the possibility to them to connect with existing external social networks. In order to facilitate the integration of these connectors into Go-myLife, we offer an External Social Networks API to allow the communication between the platforms.

During WP2, we learnt the importance of the local community for older people. In the Local Life section, we provide a space for local communities. Go-myLife users can join these and interact with the rest of the members. Though the Community Integration API, these local associations can create their own communities, finding a new way to reach their users.

Finally, in future prototypes of Go-myLife a Data Analysis API will be offered to analysis the data, obtain statistics anonymously and detect trends to be offered through a Report Dashboard for 3<sup>rd</sup> parties, allowing the performance of market analysis, as well as providing useful information on the usage of the Go-myLife platform and its various aspects.

This first prototype will be validated during the trials. Using the feedback and the new set of technical requirements, the second phase of the project will be implemented by reviewing the architecture, redesigning the APIs and creating the final version of Go-myLife.

Annex 1 Creative Commons images' references for the photo and author information.



Figure 8: My basic profile info (Mobile)

- Edit profile picture: option to upload profile photo and set as user's avatar.



Figure 9: Profile picture manager (Desktop)



**Figure 10: Profile picture manager (Mobile)**

- Login/Logout Facebook: we can login to and logout from Facebook from Go-myLife to post news and photos on this social network too. Figure 7 and Figure 8 shows the login button and the message that shows whether the user is connected to Facebook or not. When this button is pressed a new window will open to allow the user to fill in their Facebook account data.
- Mobile
  - Show user location in a map: The user can see his/her location on a map. If the user allows sharing the actual location when he/she logs into the social network the last updated position will be shown, if not, Go-myLife won't update the location of the user and the last saved position is going to appear on the map.



Figure 11: User location (Mobile)

- Desktop
  - Show and edit my optional profile information: personal information about the user, such as education, background or marital status for example, can be managed in the *General Information* section inside *Me*. Information about the user's activities and interests can be provided in a dedicated section called *Activity and interest*. Also, contact info about the user is available on the *Contact Info* link.



Textsize + - Contrast ☐ on ☒ off Speech ☐ on ☒ off

Home My network Local Life Me News Messages Books Events Search Links

Me Help Go back

**Basic Information**

General information

Profile picture

Activity and interest

Contact info

Gender	My character
<input checked="" type="radio"/> Male <input type="radio"/> Female	Adventurous
Grandchildren	Children
No	Yes
Education	Profession
Higher education	Don't show
Marital Status	Physical status
Widowed	I need a rollator
Ethnic Origin	Religion
South European	Roman Catholic
About me	
	Save

My groups

My news

My media

My events

Facebook/Twitter

Figure 12: General information form (Desktop)



Text size + - Contrast on off Speech on off

Home My network Local life Me News Messages Photo Events Search Links

Me Help Go back

Basic Information  
General information  
Profile picture  
Activity and interest  
Contact Info

home town Milano I live in the province of Lombardia postcode 3246546 Save

My groups  
My news  
My media  
My events  
Facebook/Twitter

Figure 13: Contact info form (Desktop)

- Change password: the user can change his or her password, if required.




Figure 14: Change user password (Desktop)

**User:** the user can view information about other users or friends when he or she clicks on the “view” button next to that person from anywhere inside the Go-myLife social network.

- Common functionalities (Mobile and Desktop)
  - View user basic information: The user can see the basic details (user name, full name, email) of a friend or other user

Textsize + - Contrast ☐ on ☒ off Speech ☐ on ☒ off

 Home  Local Life  News  Photos  Events  Links  English   
 Danish

User  Help  Go back

**Basic Information**  
General information  
Profile picture  
Activity and interest  
Contact Info

user name  
jals

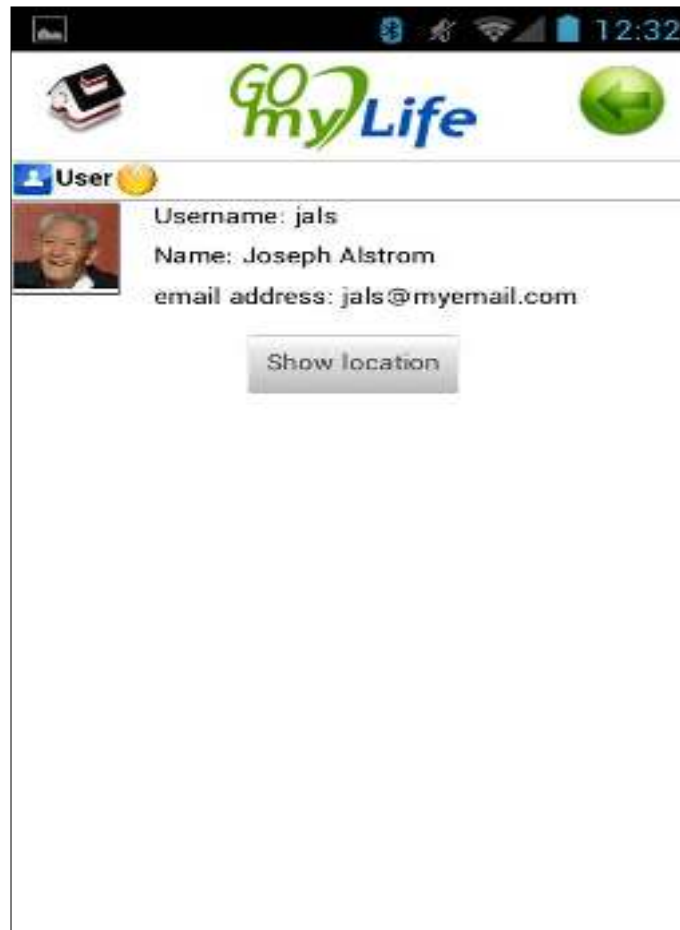
first name  
Joseph

email address  
jals@myemail.com

last name  
Alstrom



Figure 15: User basic info (Desktop)



**Figure 16: User basic info (Mobile)**

- Mobile
  - Show user location in a map or send a friend request: Just as in the *me* section, the user can see a friend's or other user's location on a map. If that user or friend allows sharing the actual location when he/she logs into the social network the last updated position will be shown, if not, Go-myLife won't update the position of the user and the last saved position is going to appear on the map. Apart from that, the user can send a request to a user that is not their friend. This functionality screen is similar to Figure 16 but with the "send request" button.
- Desktop
  - View user profile optional info of the user: the user can see contact info, general information and profile picture of a friend/other user. The screens are similar to the ones from *me* section (Figure 9, Figure 12, **Error! Reference source not found.**, Figure 13) but in this case the user only can see information and not edit it.

**My network:** section where the user will be able to arrange their contacts in groups such as close family, friends, neighbours, etc. The rest of the content of the application will be

displayed within these groups. The *All friends/All* group is always provided and is non-removable. There is another non-removable *requests* group on the Desktop version to allow the user to see friendship invitations.

- Common functionalities (Mobile and Desktop)
  - Select group: when the user clicks on “my network”, a list of groups appears on the screen. As mentioned before, “All friends/All” is always shown.



Figure 17: My network group selection (Desktop)



Figure 18: My network group selection (Mobile)

- View group of users: when one clicks on a group, a list of the users in it is shown.

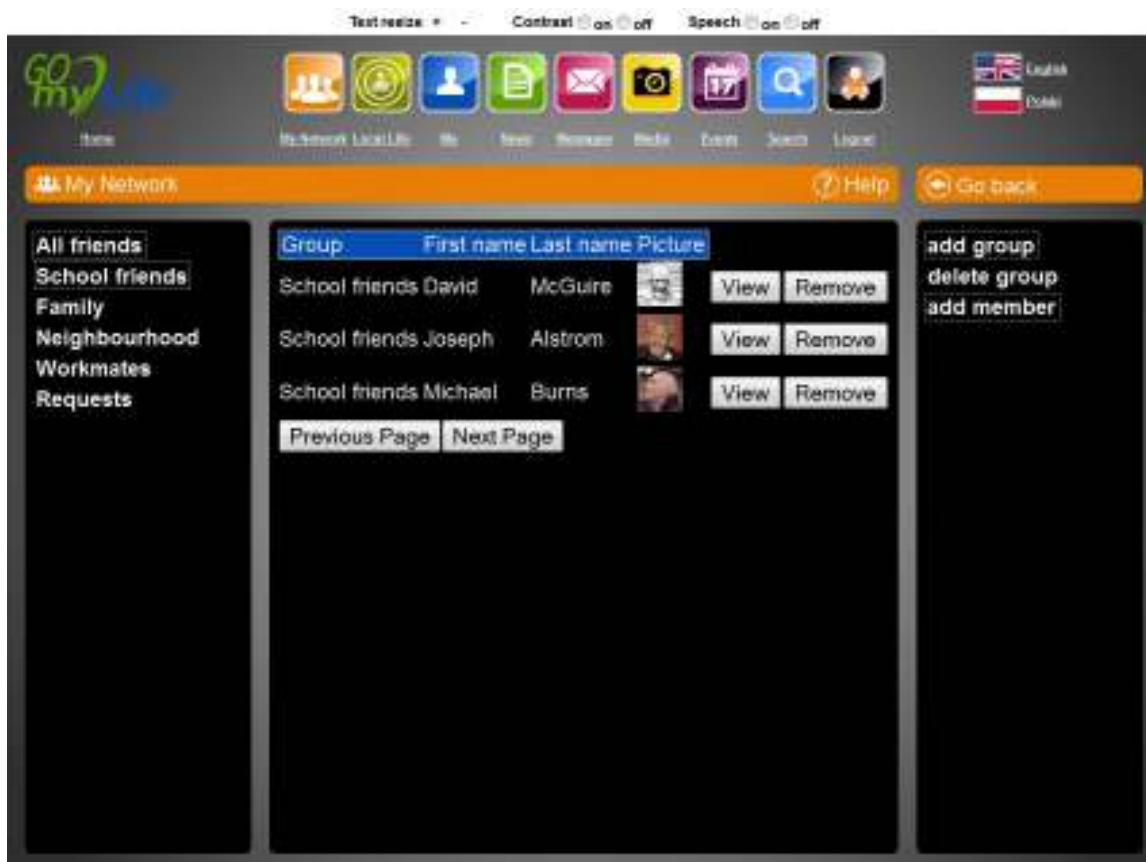


Figure 19: User group members (Desktop)





Figure 20: User group members (Mobile)

- View all user friends: when the user clicks on “All friends” group, a list of all friends is shown.



Figure 21: User friends (Desktop)



Figure 22: User friends (Mobile)

- Show friendship invitations: the user can see pending friendship invitations and accept or reject them. In the Desktop version of Go-myLife the request shows the number of common friends.



**Figure 23: Friendship invitations (Desktop)**

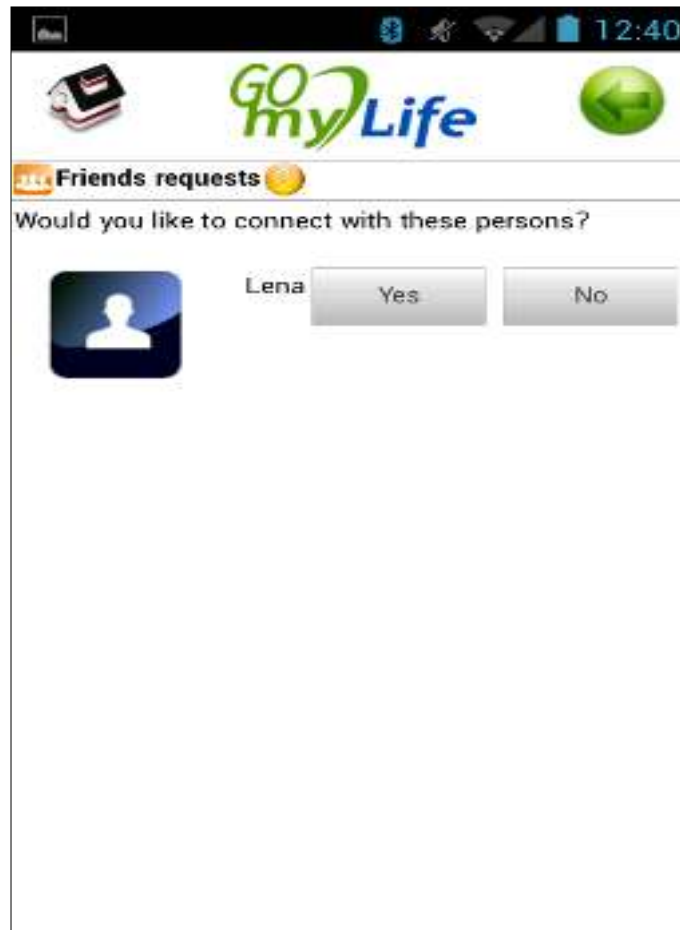
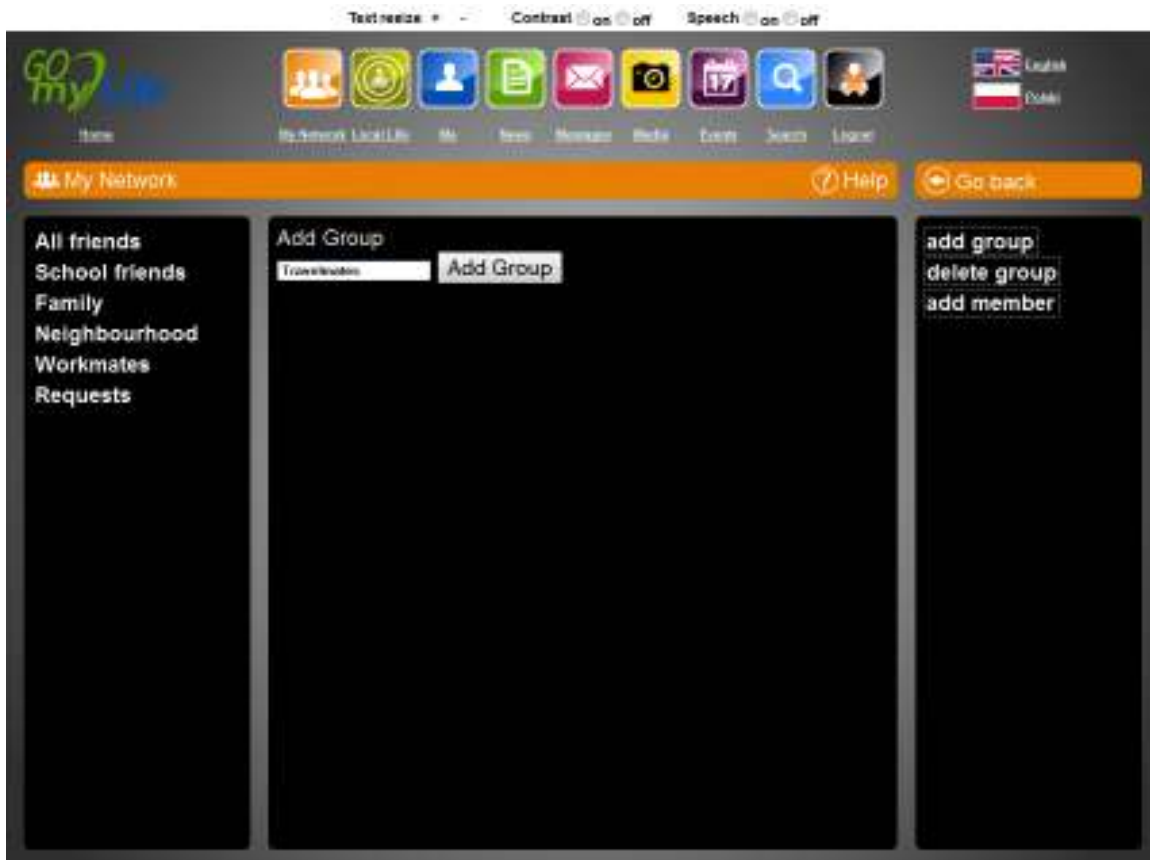


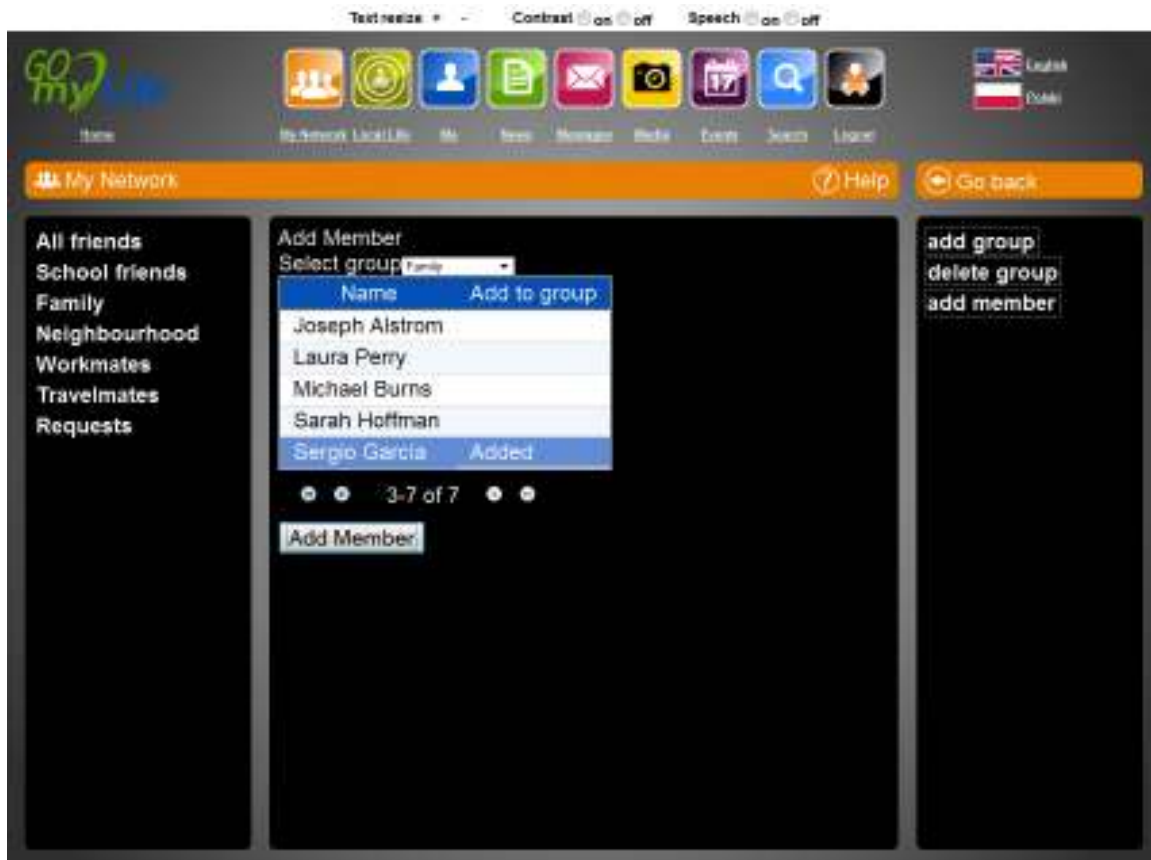
Figure 24: Friendship invitations (Mobile)

- Desktop
  - Delete users from a group: in the computer version, the user can delete users from a group with the “remove” button on the user’s list panel, as seen in Figure 19.
  - Create new user groups: in order to add a group, a user clicks on the corresponding link. Then, the central panel shows a simple form where a name for the new group is requested.



**Figure 25: The form for adding groups (Desktop)**

- Add members to a group: when a group is created, a user can add members to it by clicking on the *add member* link. The page will then allow the user to select the group to add members to and select friends to add as members of the group.



**Figure 26: The form to add members into a group (Desktop)**

- Delete group: of course, groups can also be deleted by clicking in the *delete group* options and selecting the groups to delete.



**Figure 27: Delete group list (Desktop)**

**News:** section where the user shares notes with their friends and will be able to view what their friends say and post comments to their friends' notes.

- Common functionalities (Mobile and Desktop)
  - View notes shared by friends/group of friends and share news with them: news shared by friends is shown on the screen and the user can post new notes and share them with friends.



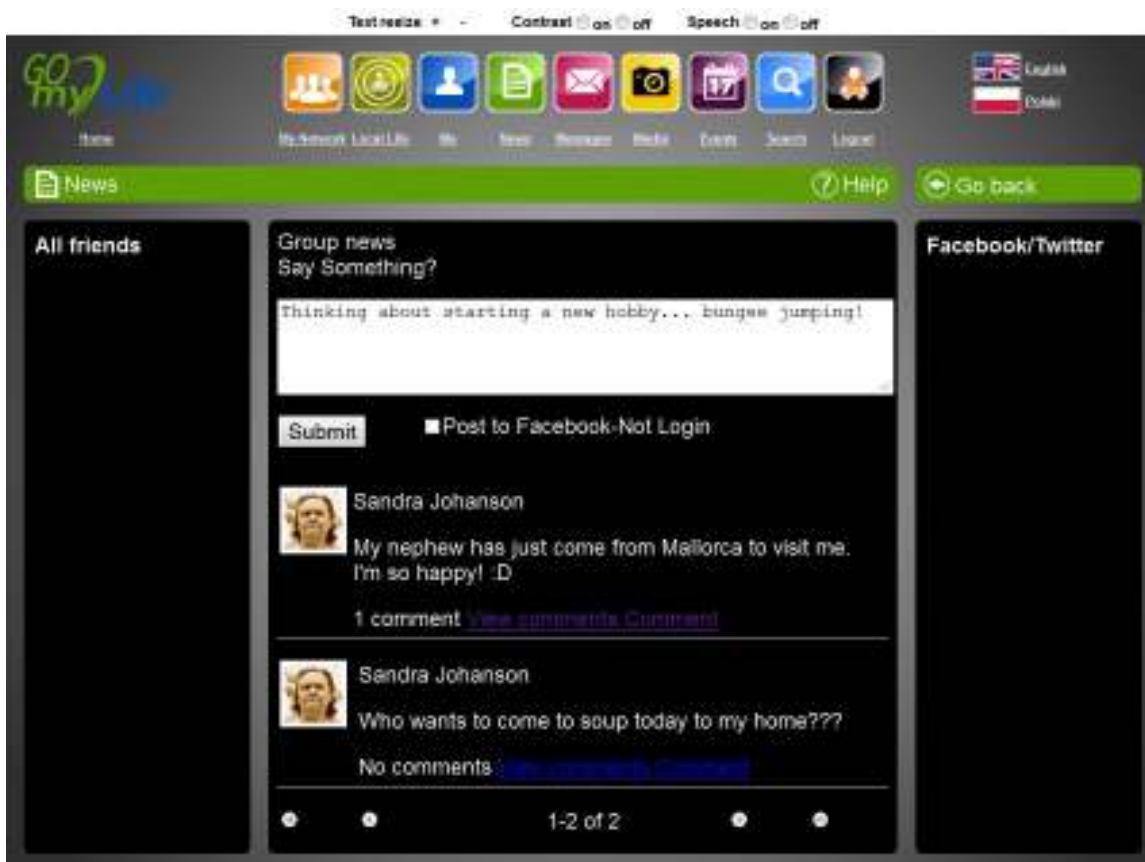
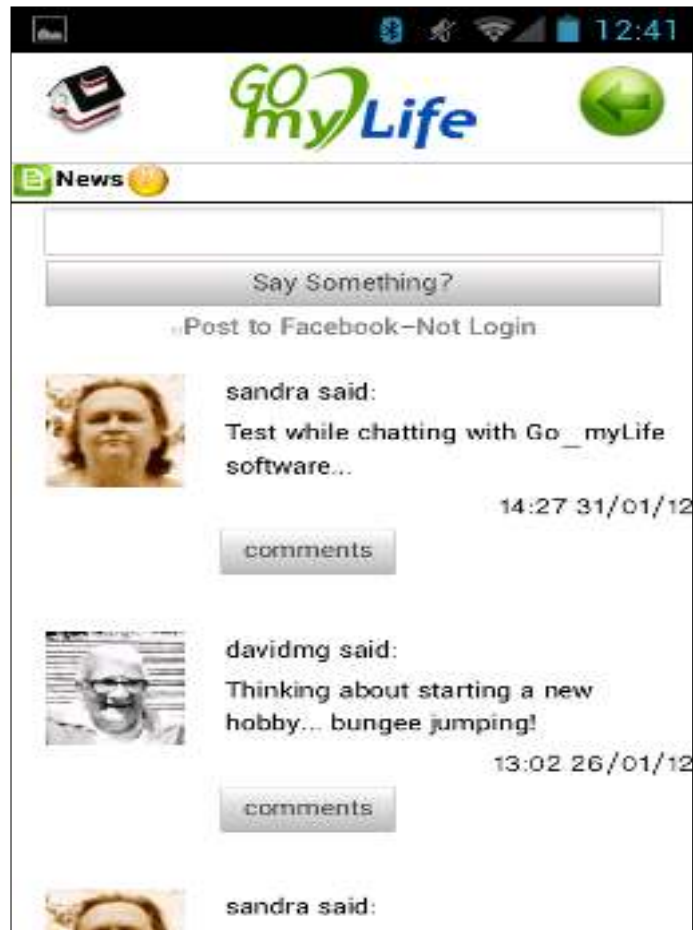


Figure 28: The News screen, writing a post (Desktop)



**Figure 29: The News screen, writing a post (Mobile)**

- Post on Facebook: the user can post the news he or she wants to share to Facebook by clicking on the “post to Facebook” checkbox. This requires the user to have previously set their status to being logged in to Facebook in the “me” section. In the previous two images, this checkbox appears under the text box. When the user enters on Facebook, the post will appear there.
- View post details and post a comment on it: by clicking on *view comments* from the Desktop or on *view* from the mobile device, details of the post appear: the post itself, date, comments, along with the photo and details of the writer of the post. The user has the option to post a new comment on the post.

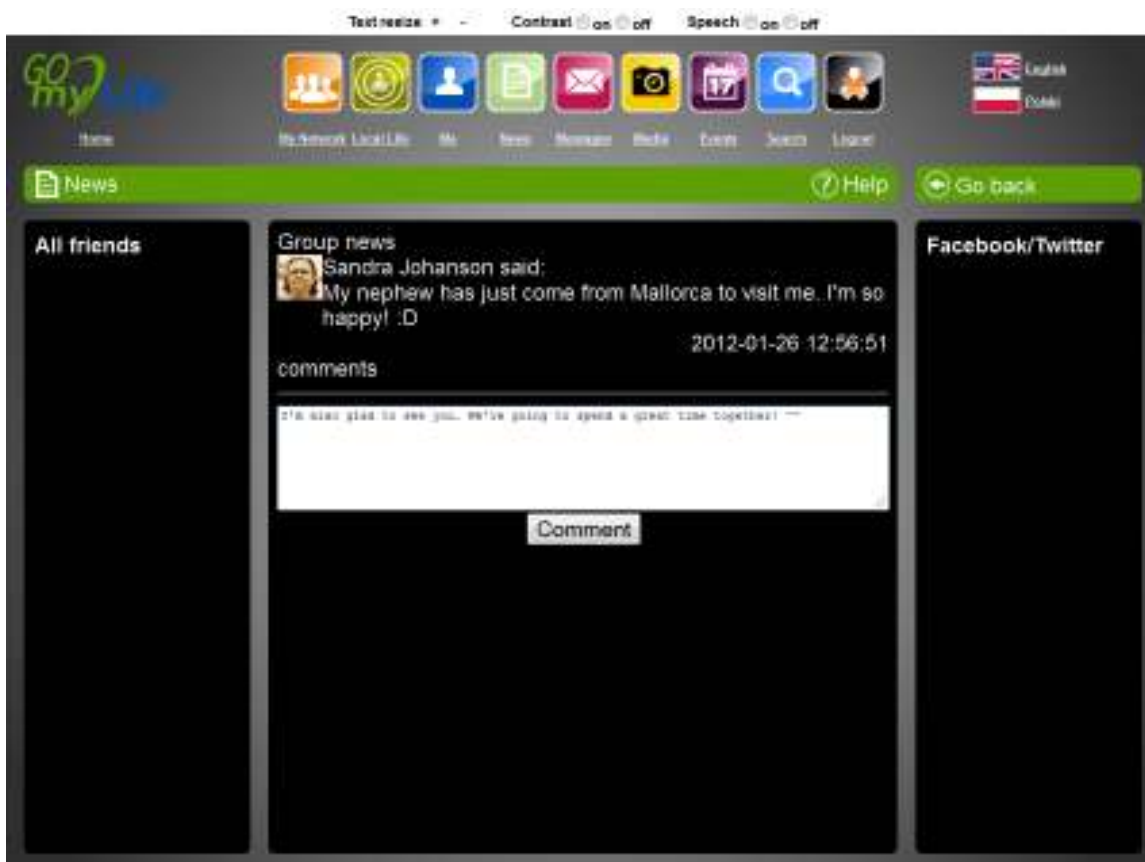


Figure 30: Posting a comment to a piece of news (Desktop)

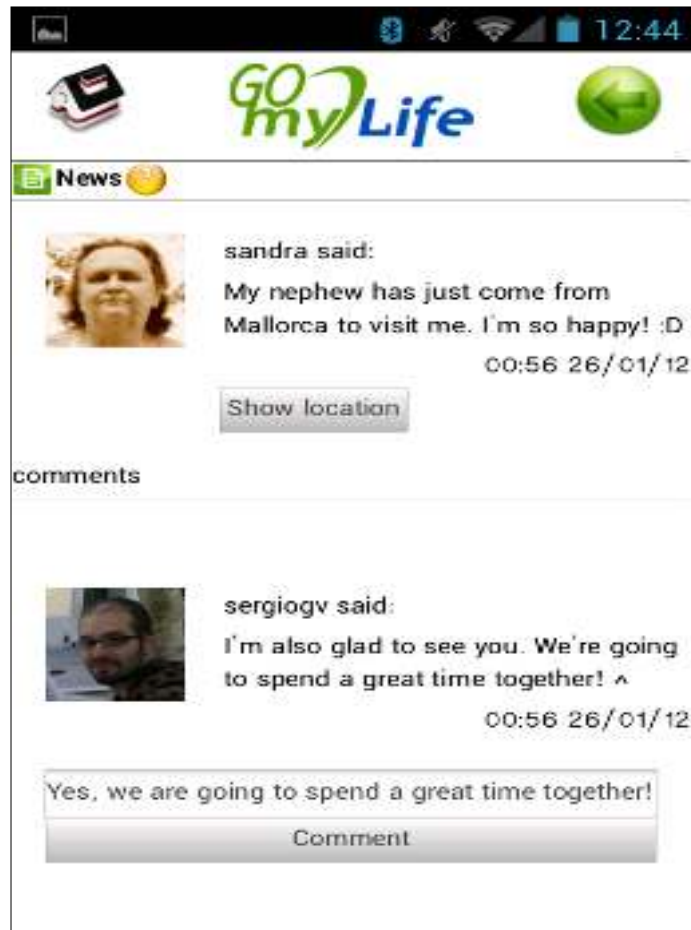


Figure 31: Posting a comment to a piece of news (Mobile)

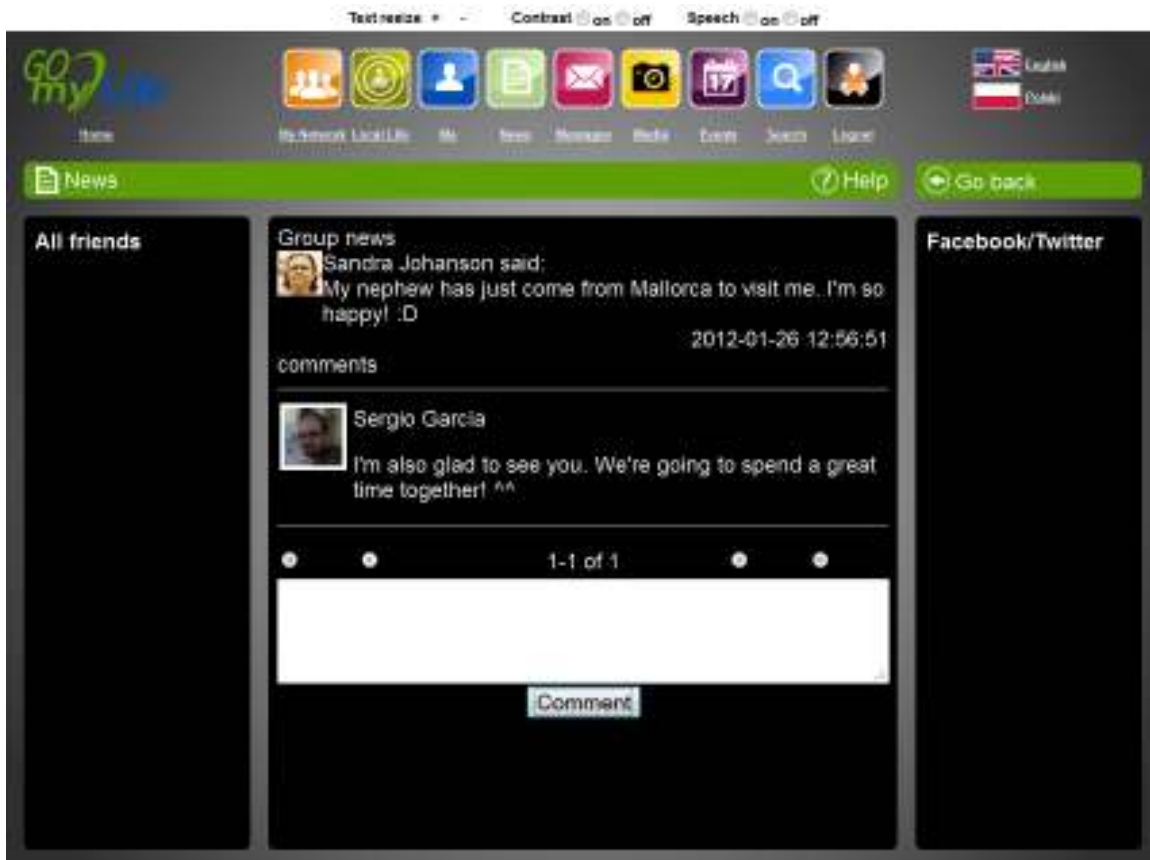


Figure 32: A commented piece of news (Desktop)

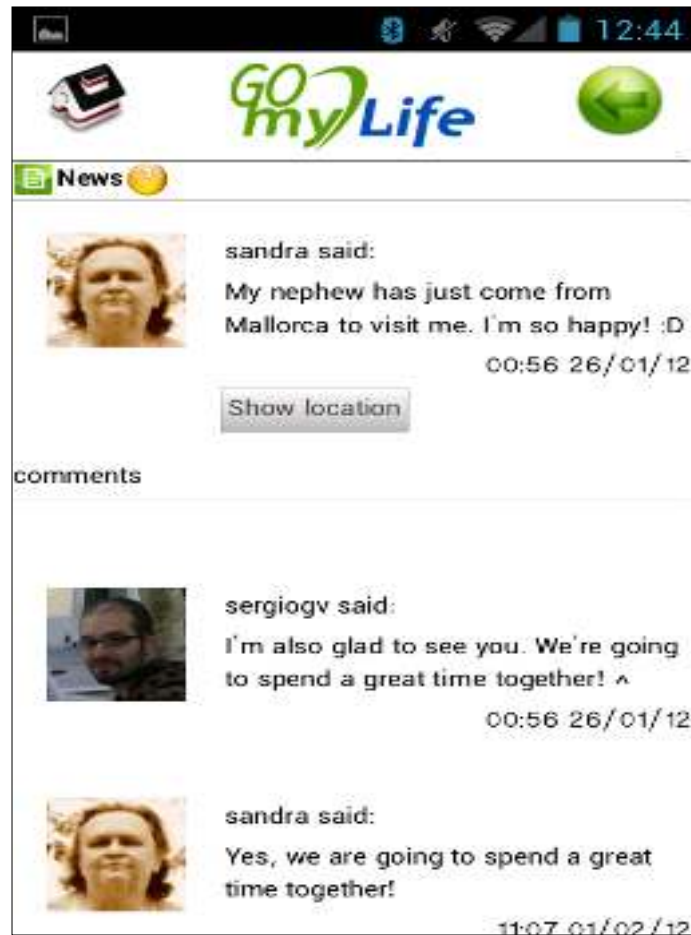


Figure 33: A commented piece of news (Mobile)

- Mobile
  - View post location on a map: similar functionality to user location that shows a map with the location of the post.
- Desktop
  - View the number of comments of the post: as is shown in Figure 28, in the desktop version of Go-myLife, the number of comments on a post appears.

**Media:** section where the user shares photos with their friends and will be able to see photos shared by his or her friends.

- Common functionalities (Mobile and Desktop)
  - View photos shared by friends/group of friends: photos shared by friends and the user are shown on the screen in thumbnails mode.



Figure 34: Main media screen (Desktop)





Figure 35: Media gallery screen (Mobile)

- Upload a photo: when the user wants to share a photo with his or her friends, he or she clicks on “upload” and a new window appears with file selection, photo title and description fields.



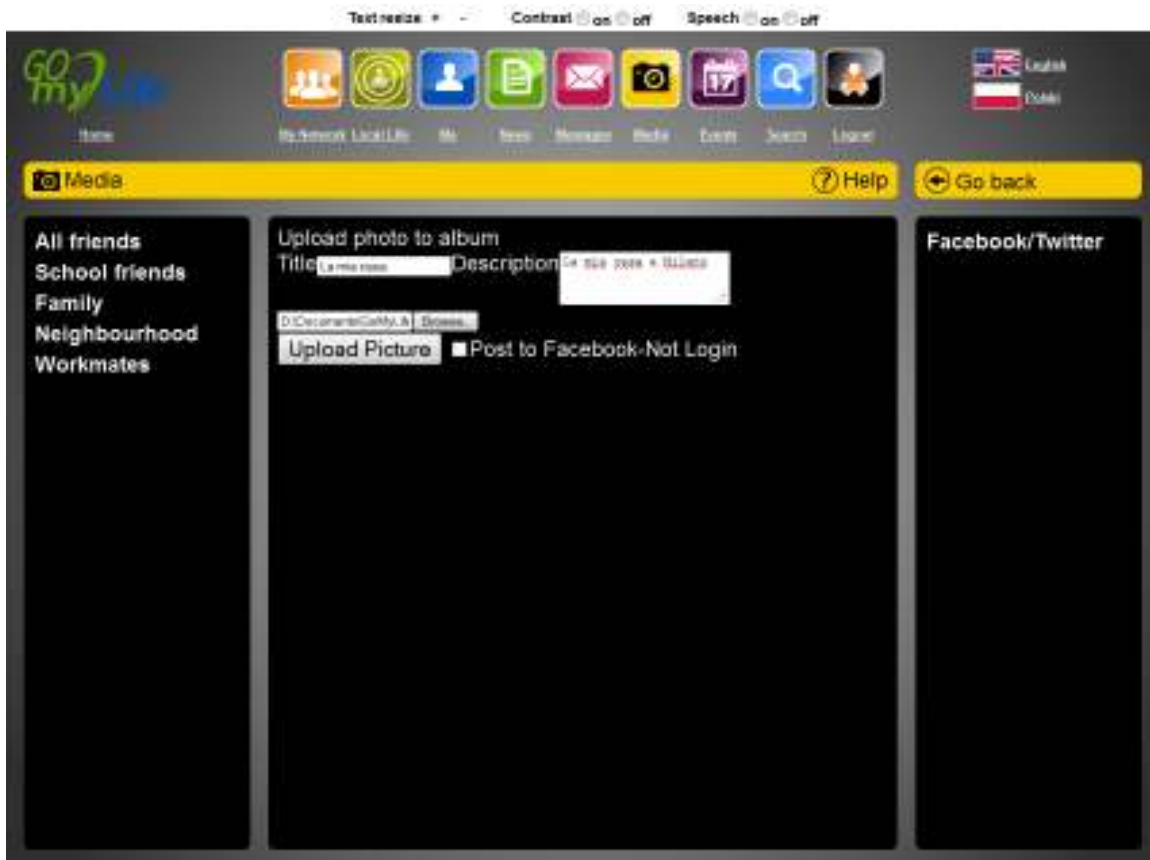


Figure 36: The Upload Photo screen (Desktop)



**Figure 37: The Upload Photo screen (Mobile)**

- Upload a photo to Facebook: as in the news section, when the user wants to share a photo and post it to Facebook, he or she will check the Facebook checkbox showed in the previous figure, click the button and the image will appear when the user logs onto Facebook.
- View photo larger: option to see details of the photo. In the mobile version the user will click on “view” to see the photo larger. In the PC version of Go-myLife, clicking on a photo shows it larger in a new pop-up, with arrows to navigate around the collection Facebook-style.

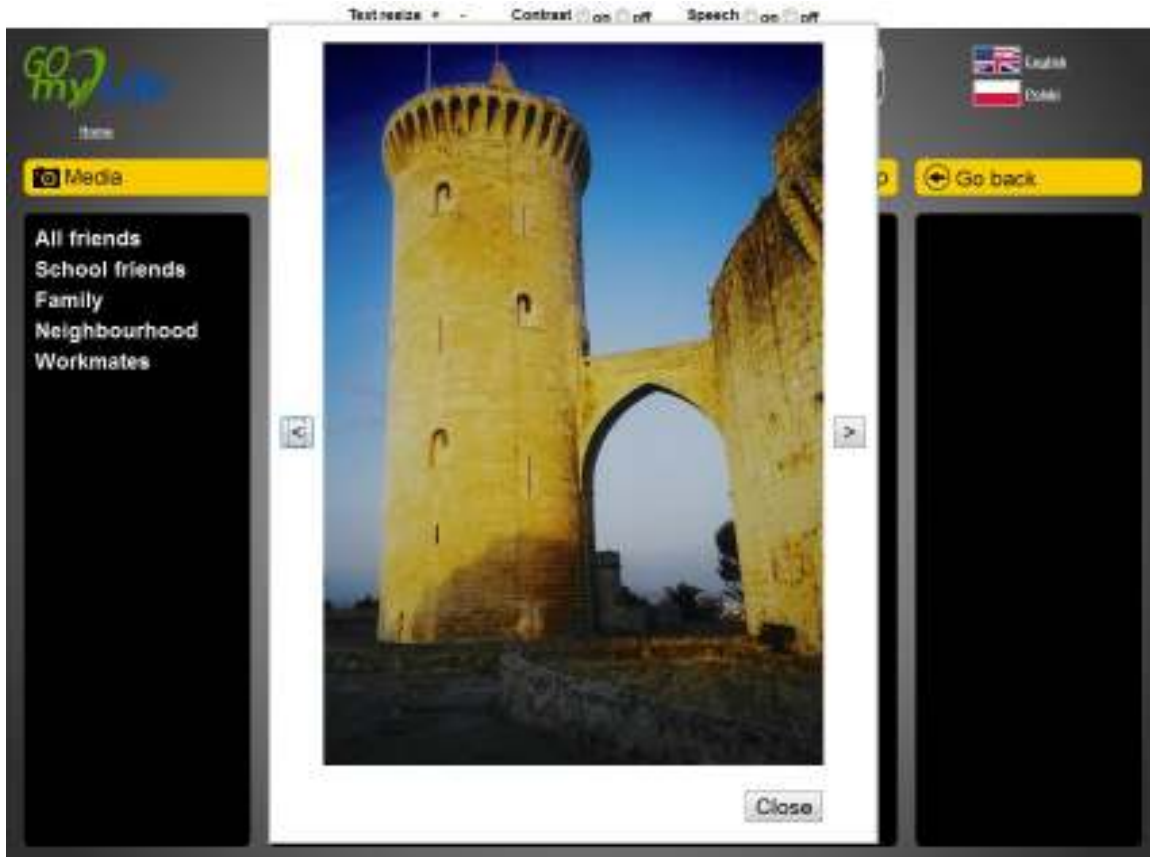


Figure 38: View of a photo (Desktop)

- Mobile
  - View photo details: apart from the option to see photo larger, photo info, uploader's name, uploader's profile photo, photo comments and the location of the photo on a map can be shown.



**Figure 39: View of a photo details (Mobile)**

- Post a comment on a photo: this functionality is similar to the one that appears on the news section. The user can post a comment on a photo from the mobile.



**Figure 40: View of a photo comments (Mobile)**

- View photos of Panoramio: the user will see a group called “Panoramio” that shows photos around the user of this network based on their location.



Figure 41: View Panoramio photos (Mobile)

**Events:** section where the user views events they have been invited to and creates their own events.

- Common functionalities (Mobile and Desktop)
  - View events shared by friends/group of friends: this section shows a list of events shared by the user in the case of mobile version and a list of accepted events in the case of Go-myLife computer version.

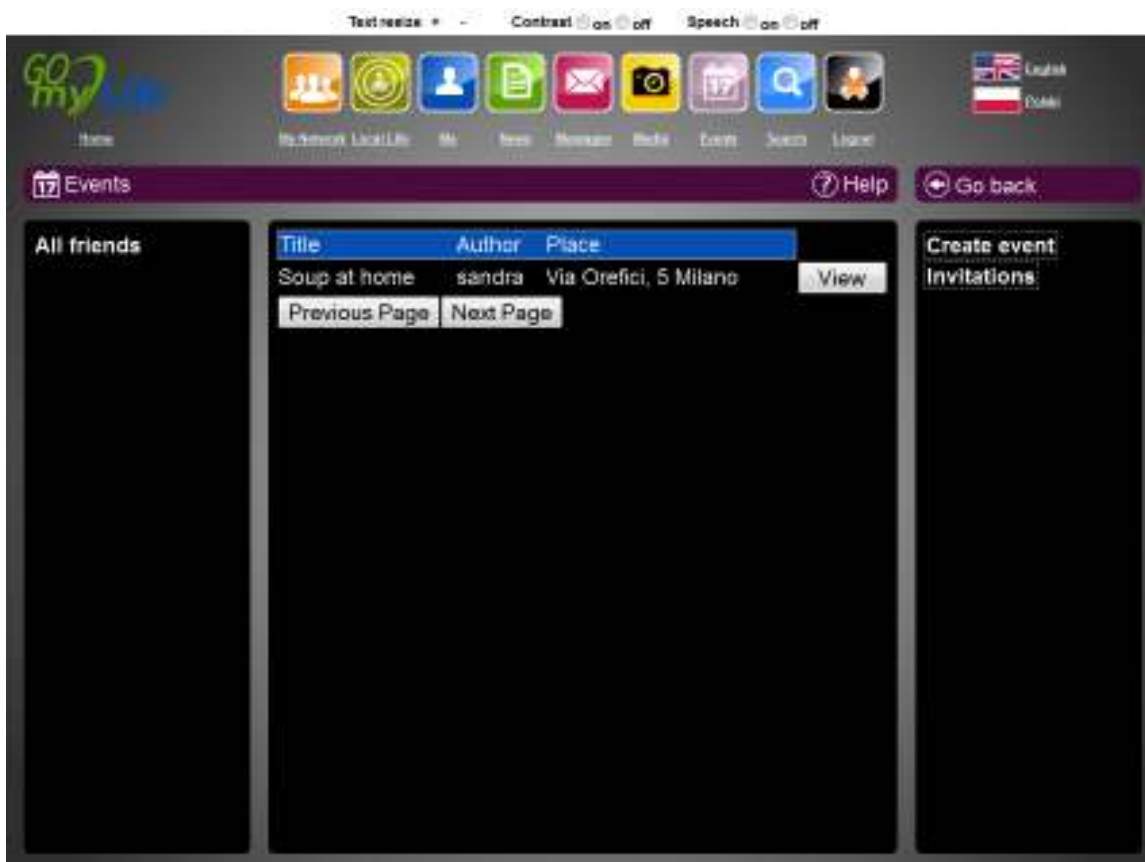


Figure 42: List of events (Desktop)



**Figure 43: List of events (Mobile)**

- View pending events invitations: this functionality shows a list of events (as in previous images (Figure 42 and Figure 43), but in this case, it shows events the user has been invited to but not yet accepted. There is a link in the event section to see pending invitations on the desktop version and a button before the list of events on the mobile version.
- View event details: in this screen the user can see the name, place, date, and description and change their response to the event (attending, not attending, maybe attending). If the event is still pending, the user can choose the relevant option. Otherwise, if the user had previously responded, they can change their option here.



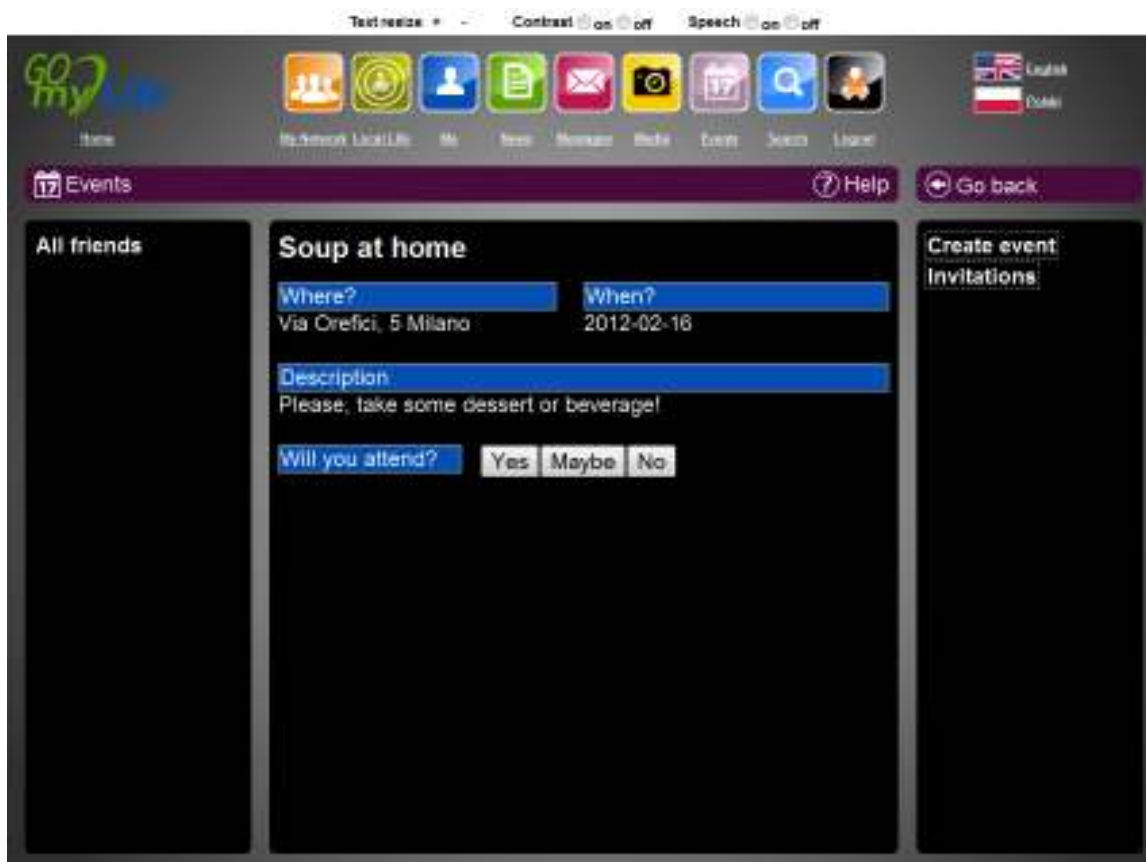


Figure 44: Event details (Desktop)



Figure 45: Event details (Mobile)

- Mobile
  - Show members response to the event: in the mobile, there is the option to see who has been invited, who will attend, who is not going to attend and who will maybe attend the event.

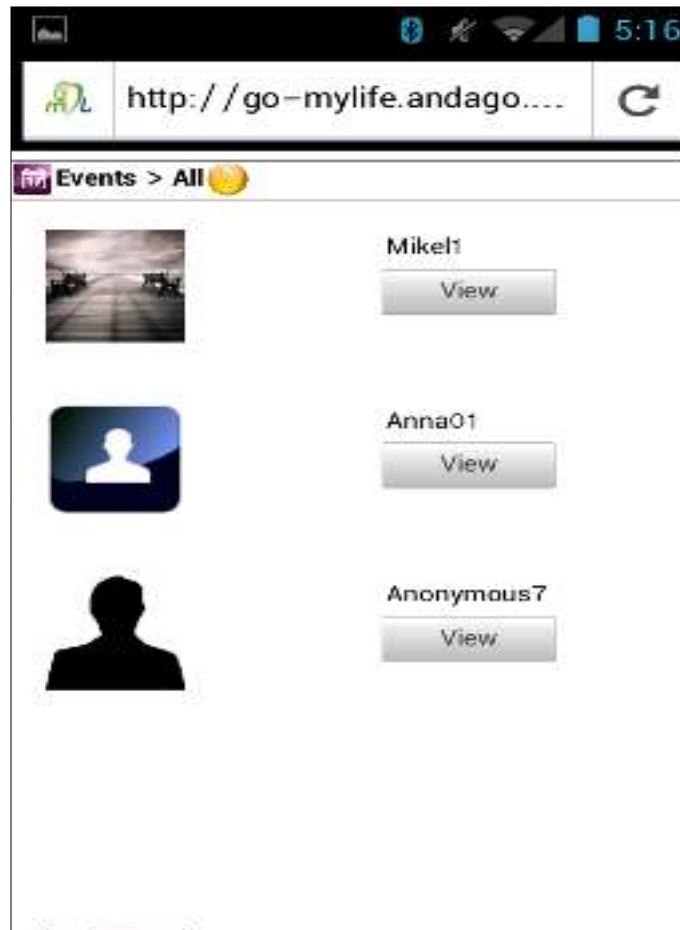


Figure 46: Users invited to the event (Mobile)

- Desktop
  - Create a new event and invite friends: in order to create a new event, the user needs the date, place, description, event name and a list of friends.

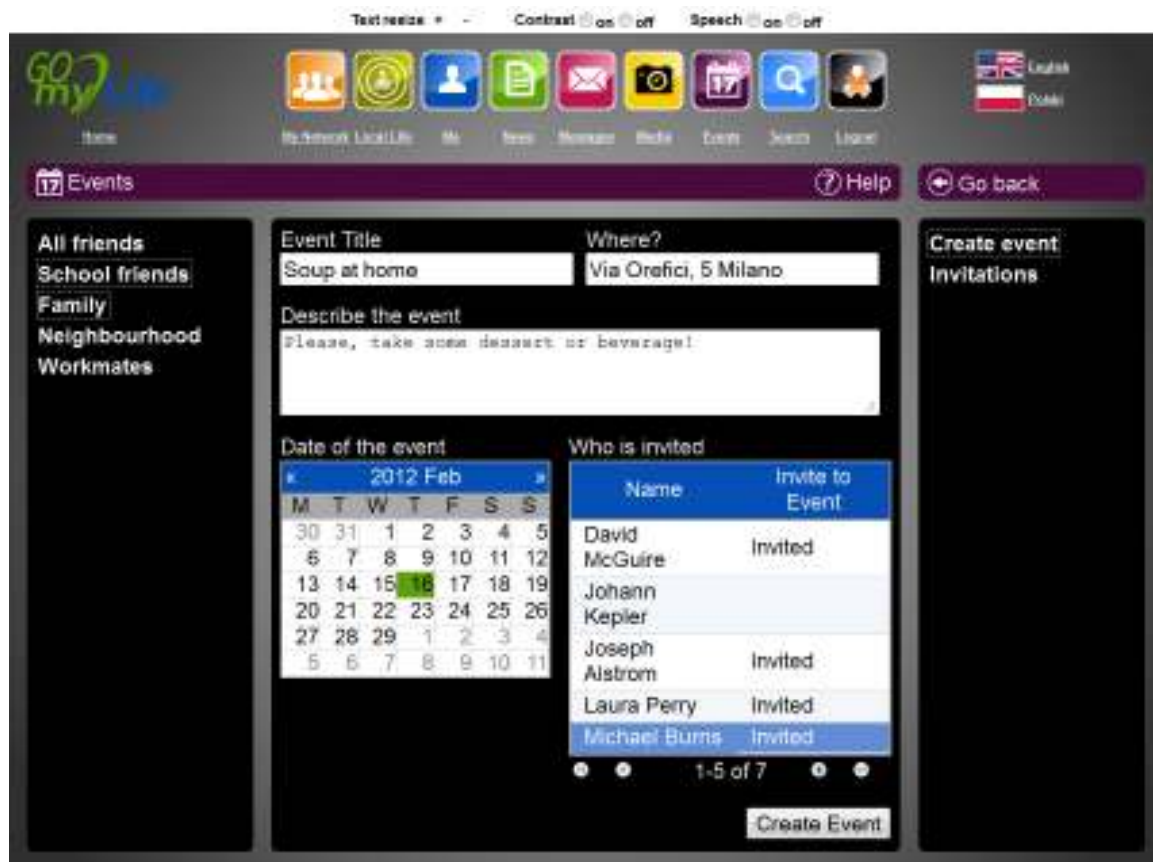
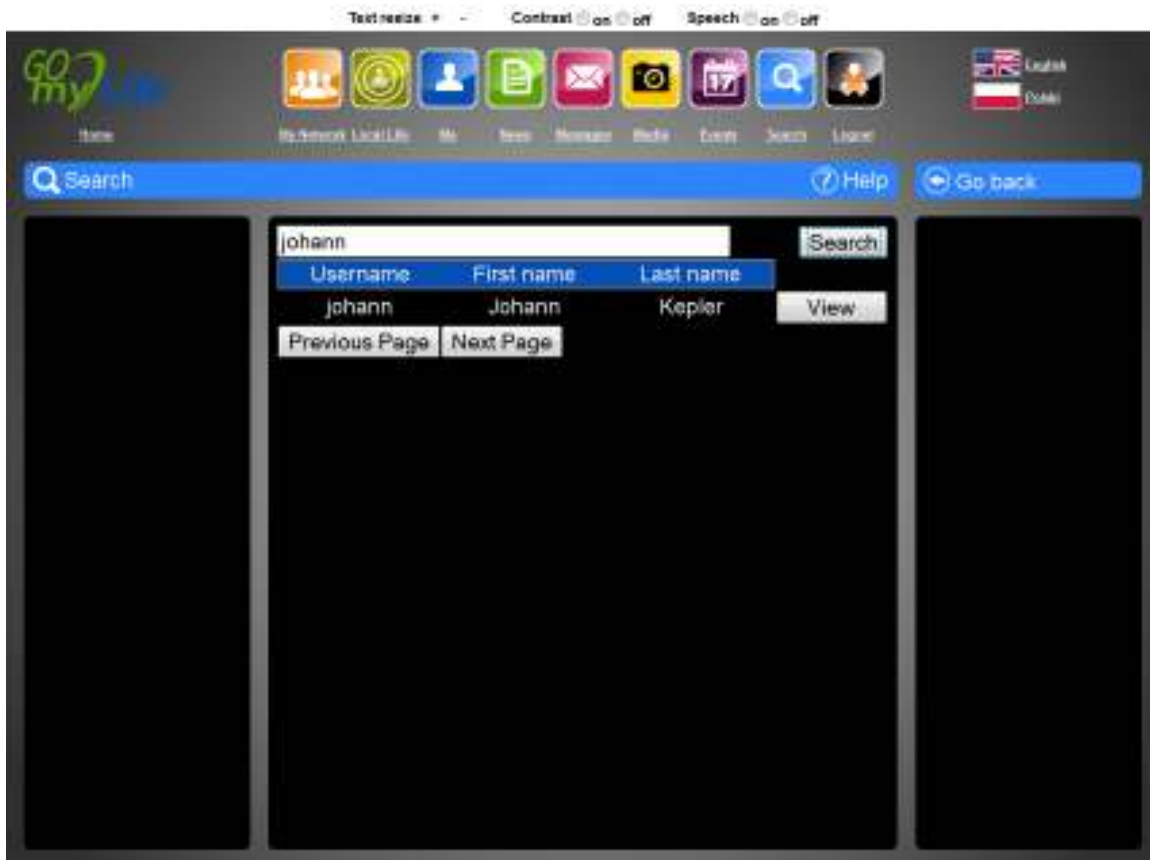


Figure 47: The screen for creating events (Desktop)

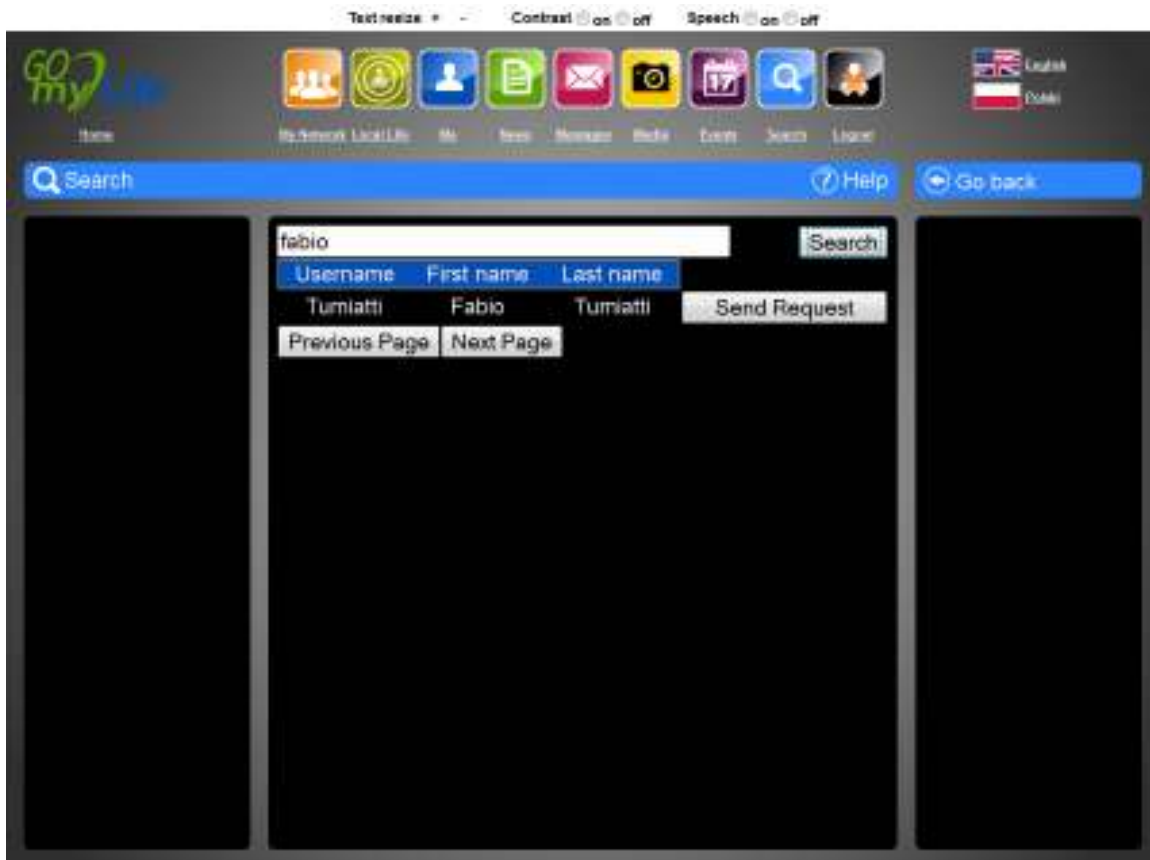
**Search:** section where the user can search people or friends within Go-myLife

- Desktop (this section and functionalities applies only to this device)
  - Search users using user name, first name or last name: this functionality offers a very simple search tool, composed only of a text field and a button. The results are shown in a simple list.



**Figure 48: Search results (Desktop)**

- View friends data if they appear in results: if a friend appears in the result query, the user will have the option to see his or her personal information with the “view” button that the previous image shows.
- Send friend request to selected user: a button to add a person as the user’s friends list is presented in the list if he/she is not already their friend, improving the ease of use of the platform.



**Figure 49: Looking for a non-friend and sending request (Desktop)**

**Help:** this part of Go-myLife guides the user during navigation inside each section of the social network

- Common functionalities (Mobile and Desktop)
  - Help text (steps) is located inside each section of Go-MyLife: In order to ensure ease of use the project has tried to ensure that the learning curve is as gentle as possible, and the steps and processes are similar to other social networks in order to help those users that have used them before. However, to further support ease of use, a Help button is provided on every screen. On clicking on this, a how-to tutorial regarding the current section appears.

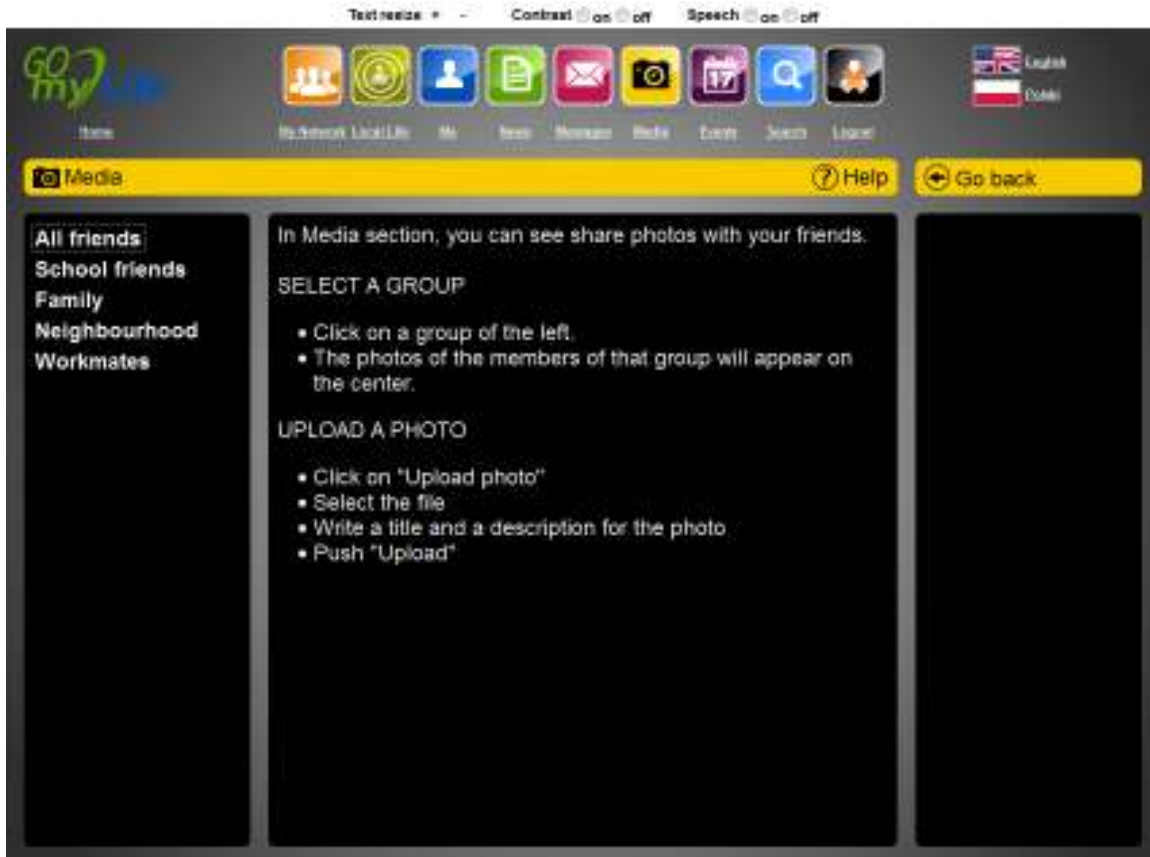
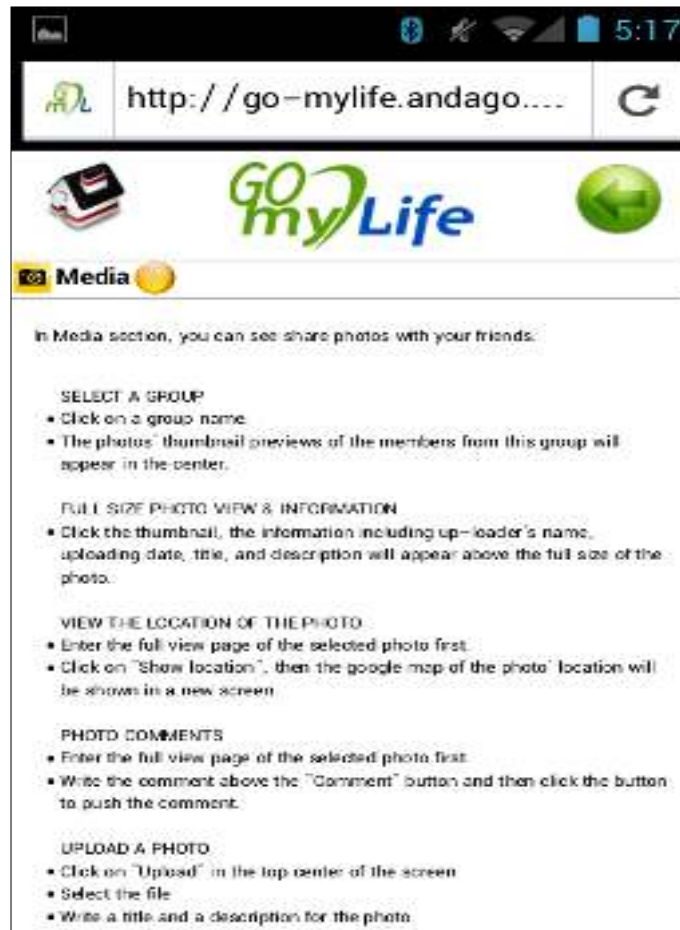


Figure 50: Help in media section (Desktop)

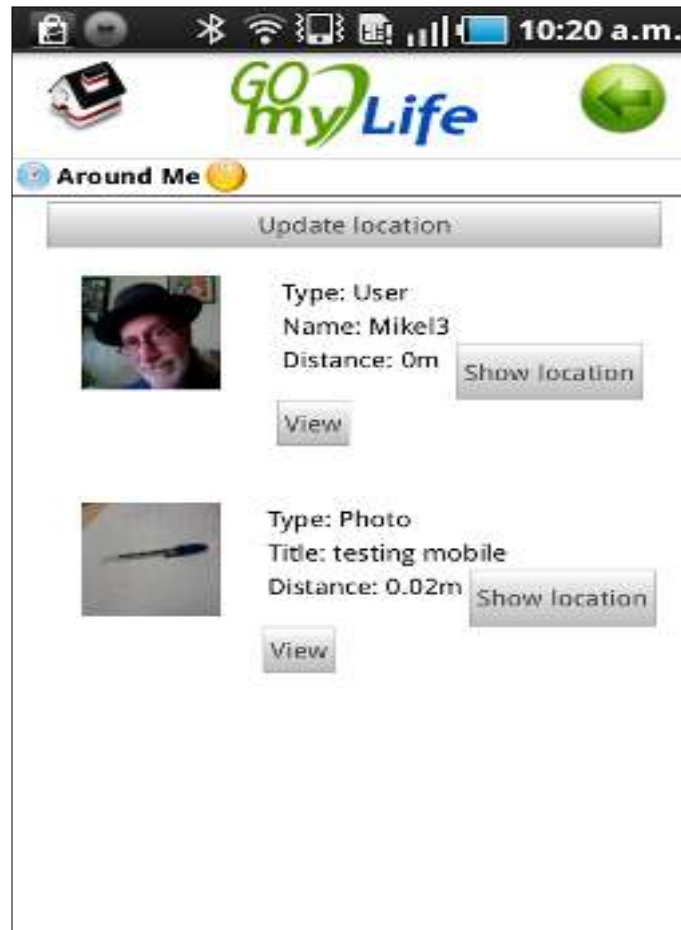


**Figure 51: Help in media section (Mobile)**

**Around Me:** section where the user can see nodes (photos, notes) and friends around his or her location

- Mobile (this section and functionalities are only offered on this device)
  - View what is around you: view nodes (photos and notes) information and friends located around the user position (radius of 5km). The nodes and friends around the user appear on a list, and the user can view details of them or see their location on a map.





**Figure 52: Around Me (Mobile)**

- Update user position: update user position on the mobile version to update and see what is around them. It can be done with “Update location” button (Figure 34).

**Language:** the user has the option to change the language of the social network and see content information in the selected language.

- Common functionalities (Mobile and Desktop)
  - Change language between Polish and English in login panel: change language by clicking on the flags of the login section. See Figure 2 and Figure 3.
- Desktop
  - Change language between Polish and English anywhere inside the social network: from the desktop version of Go-myLife the user can change the language of the social network from everywhere inside Go-myLife. In mobile, due to the limited size of the screen, this option is only in the login page.



**Figure 53: Polish interface and language flags located in the up-right corner (Desktop)**

- Change language in the registration panel: flags are located in the registration panel of the Desktop version to allow the user to select language (Figure 6)

**Local Life:** in this section the user can access local associations/groups to enable them to participate in the social life around his or her location

- Common functionalities (Mobile and Desktop)
  - View groups around the neighbourhood organized by categories: groups of people to which they can post news, photos and events. Groups can be categorized as, for instance: my neighbourhood, services, associations, politics, culture, etc.

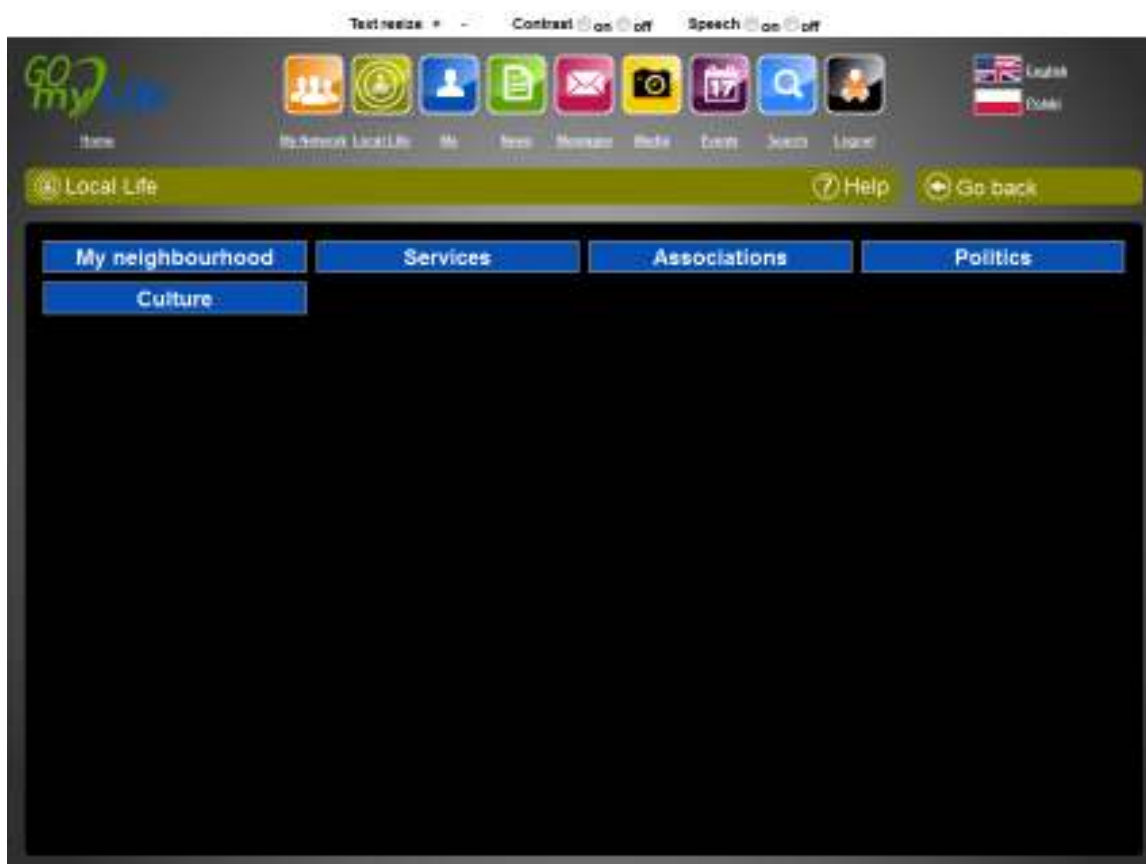


Figure 54: The Local Life main screen (Desktop)



Figure 55: The Local Life groups (Mobile)

- Join a group: if the user is not a member of the group, he/she has the option to join it.



Figure 56: Culture groups and the option to join or view them (Desktop)



**Figure 57: Join a group (Mobile)**

- Display information regarding these groups: when the user joins a group, the description of the group appears on the screen



Figure 58: Group details (Desktop)



**Figure 59: Group details (Mobile)**

- View content of groups: if the user is a member of the group, he or she can view shared photos, events, news and the list of members of the group. The screens inside this section are similar to the ones that show those types of contents in sections that have been previously described (news, photos, events, list of users). In the *members* part, the user can send a request to a user of the group that is not his or her friend when he or she see a member details.
- Upload a photo, share news and comment a post on the group: these provide the same functionalities and similar screens as with the news and media sections, to post notes, photos and comments.
- Mobile
  - Show location of user, photos, news or events of the group in a map: displays a map with the location of the node or user. (Screen with a map similar to Figure 11)
- Desktop
  - Create event on the group and invite group members: As in the events section, this functionality is only available on the desktop version of Go-myLife. The screen is similar to the *create event* panel of the Events section. (Figure 48).



## 4 Prototype design

The prototype was designed taking into account three issues: the requirements, which describe the functionality of the site and are common to all major social networks; the work done by ZSI where the preferences of older people regarding graphical user interfaces were analysed; and the capabilities and limitations of Google Web Toolkit.

Regarding the requirements, the prototype was designed to conform to the characteristics of mainstream social networks such as Facebook or Twitter. The user can recognize the commonality of the different sections for adding and organizing friends, posting, reading and commenting on news; managing and setting up groups and so on. The fact that the similarities of the user interface to other networks eases the learning curve has been a reference throughout the whole prototype design process.

The work of ZSI on the behaviour and preferences of older people when using the internet has been very important to us in this task<sup>2</sup>. The results shows some interesting conclusions; some that seem self-evident and others not. With them in mind, we have planned a site with the following characteristics:

- **Simplicity:** the main interface is composed of two main spaces, the top bar and the content. The top bar contains the main mechanisms to navigate through the site and access all its features. It is implemented using different buttons designed for clarity about which section they relate to, using both images and text. Two buttons to change the language and one to log out are also provided. This space will remain immutable in all sections, but the text will change depending on the language. This can be shown Figure 60 and Figure 61, where different sections are shown, but the top bar remains the same, unless the language is changed.

---

<sup>2</sup> All the conclusions about those studies can be found in D2.3 [Synthesis Report on target group analysis and user needs and requirements](#) and D 2.4 [Technical Requirements Analysis](#)



Figure 60: List of friends

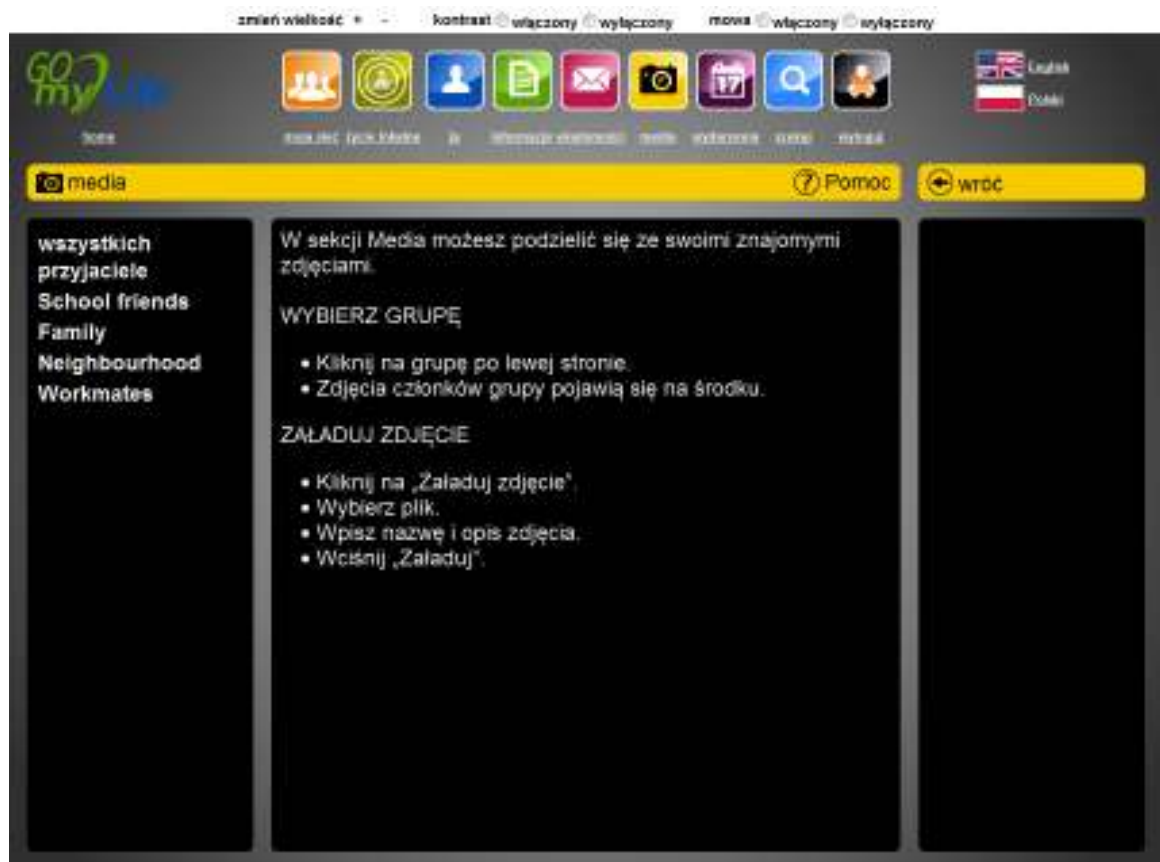


Figure 61: Help in Polish.

The content space will hold the different user information of each section. It is presented in two different layouts. In some sections, where we access information that can be organized into groups, a single panel layout filled with all of them is shown (Figure 62).



**Figure 62: My Network screen, showing the different groups where friends are classified.**

When users' information is shown, the content space will be organized in a three column layout in a thin-wide-thin distribution. The central column is where the main information will be held, while the first and last columns will contain the buttons for classifying or managing the main information. Figure 63 shows this distribution: the first column holds the classification buttons (in this case, the information is friends and the classification is done by groups), the central column shows the user data (in this case, the friends in a group), and, in the last column, the buttons to manage the information classification (in this case tools to add and remove groups, as well as a button to add a member into a group).



**Figure 63: My Network screen when the user has selected a group.**

- High contrast colours: the webpage has been designed taking into account the possible visual problems of older people. We used high contrast colours in all overlapping elements such as fonts, headers or buttons.
- Colour consistency between sections: to achieve a better navigation experience, in the content space, the colour of some elements will change depending on the section visited, matching the one in the corresponding button, as can be seen in Figure 63 and Figure 64.



Figure 64: Me section, showing a blue navigation bar, in concordance with its button.

- Large fonts as default: The entire interface has been designed to use large fonts by default. In the final prototype we plan to make them resizable.
- Help texts in every section: one of the most important points regarding ease of use is that the learning curve must not be very steep, and that similarities with other social networks will help the users that have used them before. In addition, on every screen a Help button is provided. On clicking on this, a how-to tutorial of the current section appears in the central column (Figure 65).

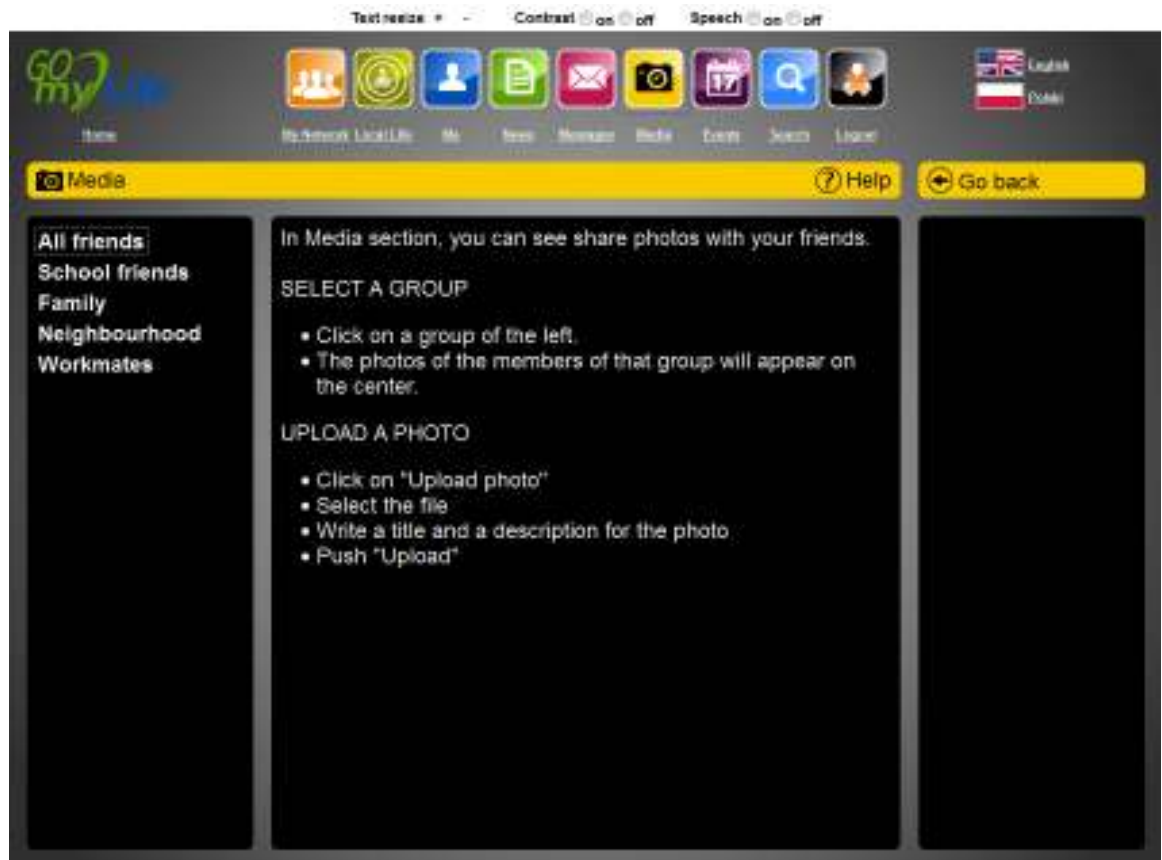


Figure 65: Help on the Media section



- No scrollbars: according to ZSI older people tend to prefer pagination when navigating in a webpage. Taking this into account, scrollable content, such as lists, has always been implemented as a set of pages, as is shown in Figure 66.



Figure 66: My Network section, showing a non-scrollable list.

- Easy completion of profile information: in order to address this, we provided separate sections when filling out profiles, classified by different aspects of the information (such as Contact Information or Activities and Interests). In some cases we also help by providing lists of choices for individual pieces of information.





Figure 67: Easy profiling.

- Easy navigation: we provide an additional “Go back” button, which is more recognisable than the one on the browser, because of the additional text description. This, in conjunction with the location bar (that provides the title of the section) and the colour coding of each section, leads to a better understanding of the navigation.

Similarly to the desktop version, the mobile prototype was designed taking into account the requirements described in D2.4 Technical Requirements Analysis, although the mobile version of the Go-myLife platform has the restrictions of the limited screen size and resolution of the mobile devices. The mobile interfaces allow constant interaction of the users with their Go-myLife account, their friends and the available services around them, using geolocation services.

In order to ensure compatibility with present and future technologies, it was decided not to attempt a native application for each and every mobile device. Instead, HTML5 code was used, implemented via the Google Web Toolkit, to enhance interoperability between devices and operating systems.

Due to the small screen size of the mobile devices and the usage of gestures and on-screen keyboards, the functionalities provided have been adapted to these specifications.

The login screen (Figure 68) allows selection between the available interface languages, password recovery and access to the help files.



**Figure 68: Mobile Login Screen**

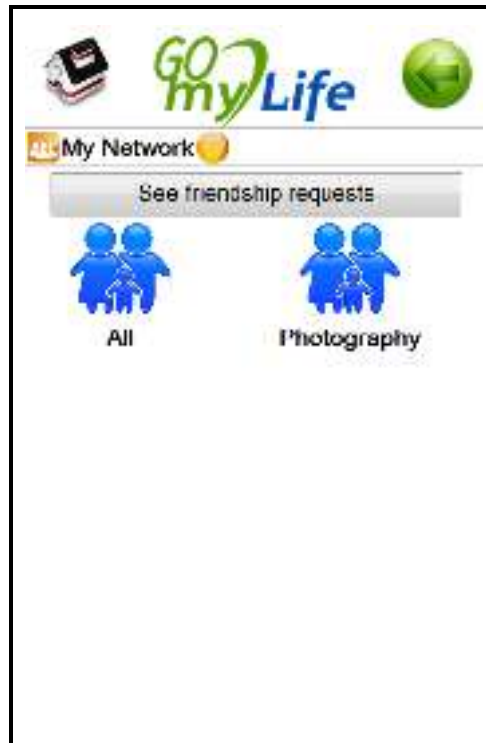
The available spaces of the screen have been reduced down to two (Figure 69). The upper one remains the same throughout the different pages, including the home button that takes the user back to the Mobile Home Screen and the back button that allows navigation back to the previous section. The only exception is displayed on the Mobile Home Screen (Figure 69), where back button has been replaced with the logout button.



**Figure 69: Mobile Home Screen**

Menu icons are the same as with the desktop version, to reduce confusion and enhance memory.

The lower space of the screen embeds the breadcrumb navigation, along with the section icon and the help button placed on top and then the contents of each selected section (Figure 70, Figure 71).



**Figure 70: My Network Mobile**

Fonts sizes used have been chosen to take into account usage by older people with vision problems. Additionally, the mobile phone screen can be zoomed in or out using gestures, to accommodate the needs of each user. Scroll bars have been avoided, although due to the limited screen size, navigating through long pages using gestures is available whenever needed (Figure 72).



Figure 71: Help Screen



Figure 72: Scrollable List

## 5 APIs, SDKs and tools

Along with the Go-myLife web applications, we provide a set of APIs to facilitate the construction of communities and the integration of external services and social networks inside the Go-myLife platform.

### 5.1.1 User interface API

The user interface API, is a REST interface to manage the content stored in the Go-myLife Social Engine. This interface is called from the UI level to create the different functionalities that the client application offers. Calls to the interface are sent via HTTP protocol and using the petitions POST and GET with JavaScript's XMLHttpRequest2.

- GET method: with this method the data of a resource is obtained. Example: the call <http://rest.gomylife.libresoft.es/social/user/data/?format=JSON> gets the user's data in a JSON file. When the response is received, the client parses it and shows the corresponding data in the user interface.
- POST method: with this method the client sends POST parameters to the URL of the resource and the REST interface modifies the DB data or creates an entry of content on it. Example: the call <https://rest.gomylife.libresoft.es/social/user/login/?format=JSON> needs POST parameters username and password. For this method, the JavaScript's FormData object is used to reproduce <form> submission mechanism. So, a file and/or a set of key/values pairs are added to this object and sent using XMLHttpRequest2. In the previous example, keys "username" and "password" with corresponding values will be added and sent to REST interface to complete login functionality.

The user interface API is divided into different modules to manage the content:

**Export management:** module that allows clients to receive the information using different data formats. The received information could be JSON or XML file, included in the HTTP Response. To obtain information in XML file, the resource URL is used, and when the client wants to receive it in JSON format, the parameter "format=JSON" will be added to the URL. With this option, when the client receives the JSON file, it parses the information and it shows it in the UI.

**Node management:** REST API module to cover standard functionalities over the different nodes (node content types: user, photo, note, video, link, audio, event):

- Change node position
- Tag a node
- Remove node tags
- Change node availability dates

**Certificates Management:** REST API module that can be used to manage security certificates of the application:

- Redirect after accepting the certificate

**User Management:** REST API module to manage user's information and relationships with other users:

- Create new user
- Modify user data (registration data)
- Modify GML profile (user optional data)
- Log in
- Reset user password
- Get user data
- Get my data
- Delete user
- Get friends
- Get friendship invitations
- Set user position
- Near people
- Near friends
- Set user status
- Set user avatar
- Unset user avatar
- List all users
- Search users
- Start relation between users
- Finish relation between users

**Notes management:** REST API module for managing notes:

- Create a new note
- Get note data
- Delete a note

- List all notes
- List all user notes

**Photos management:** REST API module for managing photos:

- Upload a photo
- Get photo data
- Get photo image
- Get photo thumbnail
- Delete a photo
- List all photos
- List all user photos

**Sounds management:** REST API module for managing sounds:

- Upload a sound
- Get sound data
- Get sound file
- Delete a sound
- List all sounds
- List all user sounds

**Videos management:** REST API module for managing videos:

- Upload a video
- Get video data
- Get video file
- Delete a video

**Links management:** REST API module for managing links:

- Create new link
- Get link data
- Delete a link

**Events management:** REST API module for managing events:



- Create a new event
- Get event data
- Delete an event
- Change the attendee's subscription

**Privacy management:** REST API module that manages the privacy of the nodes and layers of Go-myLife network. The privacy options can be: private (only the user can see it), friends (the user's friends can see it), friends of friends (the friends of the friends of the user can also see it), public (everybody).

- Show privacy status (nodes, layers)
- Permission change (node, layer)

**Layers management:** REST API module to manage information about layers. The layers manager provides an interface to access the different information sources supported in the server, allowing users and applications to interact with nodes from several sources. These sources could be internal sources (with nodes stored in LibreGeoSocial server data base) or external sources (retrieving information through web services, for example: Panoramio). Both layers (external, internal) provide the same features using a unique interface.

- Get the layers list
- Make a layer search
- Get layer's icon
- Get layer's info
- Get layer's category
- Create a layer
- Delete a layer

**Walls Management:** REST API module for managing user's walls. Internally, a wall is a layer of type wall, so the working is very similar. They use the same interface with special features. This interface is used to get user's wall or search nodes in that wall.

- Get the walls list
- Make a wall search
- Get wall's icon
- Change wall's icon
- Get wall's info

- Get wall's category
- Get user's wall

**Comments management:** REST API module that manages comments on nodes. The comments are shown in nodes data (for example, comments posted on a note).

- Add new comment
- Delete comment

### 5.1.2 Services integration API

This section explains how to develop a new external channel, in order to integrate an external information source (such as Flickr, Panoramio, YouTube, etc.) into LibreGeoSocial layers system. A channel can be developed to integrate any information source and this will remain integrated in LibreGeoSocial.

All the contents managed by LibreGeoSocial are represented inside a layer. The layers are the abstraction used in LibreGeoSocial to classify the information. When a channel is developed it will be reflected as a new layer from the point of view of LibreGeoSocial server. The difference is that the social contents belonging to this layer won't be stored in the LibreGeoSocial database, so they will be requested from their original source every time the channel is loaded.

For the creation of a channel, a new class has to be created, that inherits from the class *GIC.Channels.GenericChannel* (Figure 73):

```
class <OwnChannel> (GenericChannel):
```

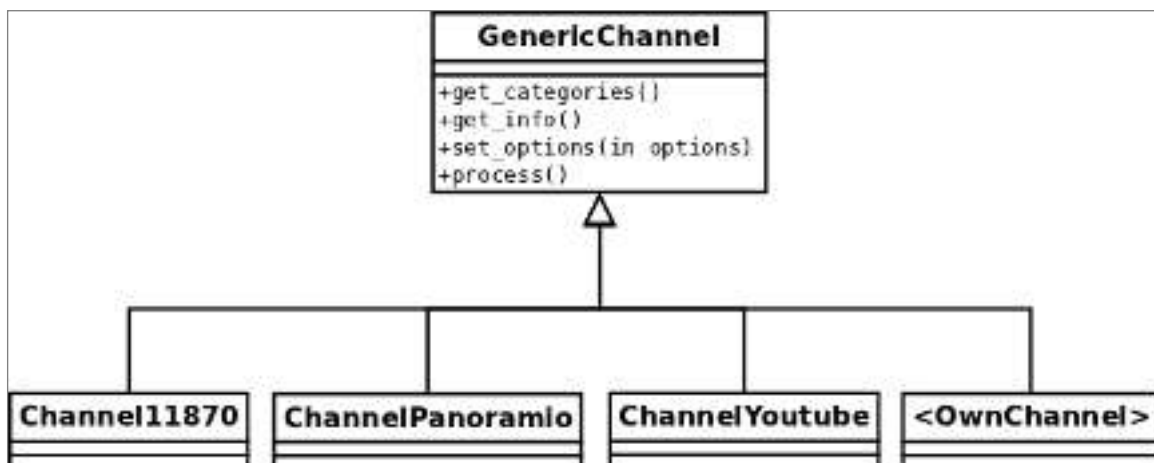


Figure 73: The External Channel Aggregator

Next, these are the methods of the API that have to be implemented:

- *get\_categories()*:

**Description:** This function has to return information about the different categories (if supported) by the channel. For example, in a video channel categories could be: “sports”, “movie”, “music”.

**Output:** returns an array of dictionaries. Each dictionary will have the next keys for each category:

- “id”: identification number
- “name”: identification name
- “description”: description of the category.

This information will be useful for users and it will be showed in the LibreGeoSocial layers description.

- ***get\_info():***

**Description:** Returns the description of the channel. This information will be showed in the description of the channel in LibreGeoSocial.

**Output:** a string with the channel description

- ***set\_options():***

**Description:** This function configures the channel regarding the different options supported. LibreGeoSocial needs a set of mandatory options and this function will receive, at least, these fields (use them as your convenience). Other mandatory and/or optional parameters could be added, but it is not recommended to accessing the channel through the LibreGeoSocial Standard Layer System. This system tries to call all the layers with the same options; so, new layers could be added to the system without needing to make changes in the client (which always makes the same request, not worrying about the requesting layer).

**Input:** a dictionary with the different options. This dictionary will be composed with pairs of options/values.

**Output:** a pair of values with True/False if the options are correct and a returned message. This returned message could be used, for example, to refer a missing mandatory field: return False, “The parameter latitude is mandatory”

- ***process():***

**Description:** This function is the core of the channel and it will make the search/request returning the matching information.

**Output:** The matched information regarding different options and the information source have to be returned regarding the LibreGeoSocial content models. Currently, the different information nodes can be returned through the LibreGeoSocial standard ones: notes, photos, audios and videos. So, this function returns a pair of values: the first value is a boolean (true/false) notifying if everything was right and the second one has to be an array of LibreGeoSocial nodes.

In order to return LibreGeoSocial nodes in a standard way, the classes designed for this purpose have to be used. These classes represent the LibreGeoSocial standard nodes that are used to return information with internal templates. Using another node's models could not fit correctly with these templates. The classes to use LibreGeoSocial standard nodes can be found in: *GIC.Channels.Items*.

Here is an example of how to return an information node in the standard layers system:

```
from GIC.Channels.Items import Note

note = Note()
note.title = "Title information"
note.text = "Text information"

return (True, [note])
```

This “simple” API would allow developers to integrate different information sources quickly.

Finally, to install a channel into a LibreGeoSocial server, a directory has to be created inside the libs directory of the LibreGeoSocial server installation with the next convention: “channel<name of the channel>”. Here is an example for a channel called Panoramio:

```
cd /var/www/libregeosocial/libs
mkdir channelPanoramio
```

Then, put the python file/template inside the directory, the file has to be named as the directory with the first letter uppercase:

```
cp /home/user/development/panoramio.py
./channelPanoramio/ChannelPanoramio.py
```

The class implemented by this file/template/channel also has to use a convention: “Channel<name of the channel>”<sup>3</sup>. For example:

```
class ChannelPanoramio(GenericChannel):
```

---

<sup>3</sup> The system is case sensitive. For the Panoramio channel the directory will be channelPanoramio and then, the class name will be ChannelPanoramio. The first letter is important, so this will be used by the layer system to detect and to instance channels automatically.

With the channel class correctly installed it will have to be inserted in the layers system. There is a python script to manage this task under *social.layers.LayerConsistency*. It should be invoked inside a Django instance:

```
$> python2.5 manage.py shell
shell> from social.layers import LayerConsistency
shell> LayerConsistency.main()
```

This script will search new channels in the system.

### 5.1.3 Communities Integration API

The Communities Integration API allows the interaction of third party providers with the Go-myLife social network. With this API, local communities or associations can create their own groups, share content with their members and manage it. The API is implemented using a REST interface and can be called over HTTP protocol, using GET and POST requests. In that sense, the administrator of the community may create, update or delete content related to a community.

To facilitate the creation and management of these local groups, a Community Admin page will be implemented. The potential third party administration must have available the access key, provided by the administrator of the Go-myLife platform upon verification, in order to access the Community Admin page.

The Communities Integration API allows content creation/update and reception of the posted content. The reception of the content is performed using a GET method, e.g. <https://rest.gomylife.libresoft.es/social/group/<groupid>/<parameters>/?format=JSON> and content creation and update using a POST method, e.g. <https://rest.gomylife.libresoft.es/social/group/<groupid>/<parameters>/?format=JSON>. Calls to the interface are sent via HTTP protocol using the petitions POST and GET with JavaScript's XmlHttpRequest2. The *groupid* defines the specific local group, while *parameter* defines the action to be performed. The available operations are:

#### Communities' management: REST API to manage communities

- Get the communities list
- Create community (name, description, category, icon, availability, latitude, longitude)
- Get community data (name, description, category, icon, creation date, latitude, longitude)
- Get community members
- Show community elements
- Make a search inside a community
- Delete a community

- List all user's communities
- Join a user to a group
- Delete a user from a group

**Community news management:** REST API to manage communities' news feeds

- Get community news list
- Send news content (title, news\_content, available\_to, available\_from, latitude, longitude, altitude)
- Delete news

**Community photos management:** REST API to manage communities' photos

- Get community photos list
- Send photo content (name, description, photo\_file, longitude, latitude, available\_to, available\_from, altitude)
- Delete photos

**Community sounds management:** REST API to manage communities' sounds

- Get community sounds list
- Send sound content (name, description, sound\_file, longitude, latitude, available\_to, available\_from, altitude)
- Delete sounds

**Community videos management:** REST API to manage communities' videos

- Get community videos list
- Send sound content (name, description, video\_file, longitude, latitude, available\_to, available\_from, altitude)
- Delete videos

**Community links management:** REST API to manage communities' links

- Get community links list
- Send links content (title, text, url, longitude, latitude, available\_to, available\_from, altitude)
- Delete links

### Community events management: REST API to manage communities' events

- Get community events list
- Send events content (title, text, url, longitude, latitude, event\_to, event\_date, available\_to, available\_from, altitude)
- Delete links

#### 5.1.4 External Social Networks API

This API provides a connection to other social networks analysed in previous deliverables. It allows a bidirectional communication between the platforms. Different connectors are developed for the interaction with each of these networks. All of them follow the next architecture schema to guarantee its integration without unnecessary changes at the user interface:

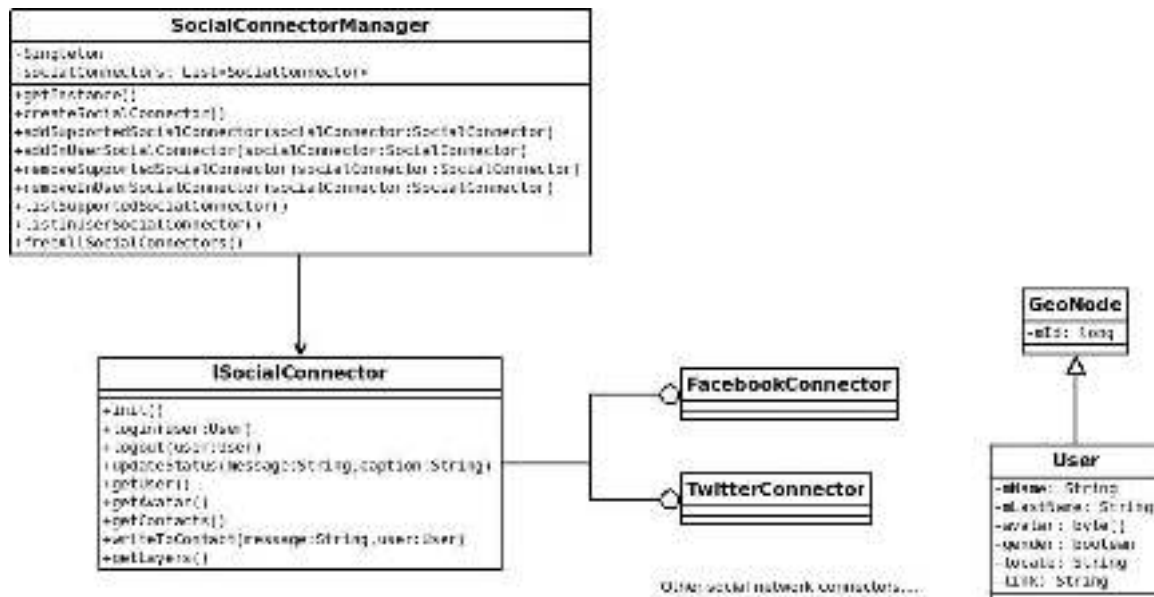


Figure 74: External Social Network API schema

The SocialConnectorManager is a factory class that creates all the instances of Social Connectors and manages them. With this API, the inclusion of new social networks in the Go-myLife platform will be easy because it is only necessary to implement the ISocialConnector interface for each social network. Implementing this interface allows the platform to obtain and post data to each specific social network, in this case, the data from/to Facebook and Twitter.

Using ISocialConnector interface, these options are possible: login/logout, update user status, get user information, get user profile picture, get user contacts, write to a contact, etc.

### 5.1.5 Data Analysis API

The Data Analysis API will provide a series of queries, available via the Report Dashboard, that allow anonymous usage and performance statistics. The Report Dashboard will offer predefined and custom queries to the certified users. Go-myLife platform reports will only be available to verified users and partners, via secure connection, so that to protect the data from non-authorized usage. Data Analysis API queries will provide statistics in relation to demographics, location, activities, communities etc.



## 6 Conclusion

This deliverable describes the first Go-myLife prototype implemented during the first phase of the project and that will be tested during the pilots in UK and Poland. The Go-myLife prototype consists of the Go-myLife Social Network Engine, a set of APIs and web-based client applications to be accessed through the desktop or the mobile.

The Go-myLife Social Engine is based on the *LibreGeoSocial* framework and is the responsible for the management of the content, its relationship and its privacy. Over it, a set of APIs have been built in order to facilitate the integration of external content and the construction of communities.

The Go-myLife Social Engine offers its functionality through a REST API (User Interface API). Making use of it, we have built the web-based applications (desktop and mobile) where the users interact and share their content. Different social functionalities are offered in this first version: manage your contacts (My Network), share posts (News), share photos (Media), organize events (Events) or update your profile information (Me).

But Go-myLife takes the social network one step further and provides information about who and what is around the user (Around me), where the end-users can find local services, friends or content uploaded by others. Through the Services Integration API, local companies or associations can integrate their services into Go-myLife in order to be found by the users.

Go-myLife also recognizes the relevance of existing social networks where people within the networks of the end-users may already have an account. Go-myLife doesn't want to create an isolated and exclusive social network for older people but it wants to offer the possibility to them to connect with existing external social networks. In order to facilitate the integration of these connectors into Go-myLife, we offer an External Social Networks API to allow the communication between the platforms.

During WP2, we learnt the importance of the local community for older people. In the Local Life section, we provide a space for local communities. Go-myLife users can join these and interact with the rest of the members. Through the Community Integration API, these local associations can create their own communities, finding a new way to reach their users.

Finally, in future prototypes of Go-myLife a Data Analysis API will be offered to analysis the data, obtain statistics anonymously and detect trends to be offered through a Report Dashboard for 3<sup>rd</sup> parties, allowing the performance of market analysis, as well as providing useful information on the usage of the Go-myLife platform and its various aspects.

This first prototype will be validated during the trials. Using the feedback and the new set of technical requirements, the second phase of the project will be implemented by reviewing the architecture, redesigning the APIs and creating the final version of Go-myLife.

## 7 Annex 1 Creative Commons images' references

Artis Rams (artisrams) (<http://www.flickr.com/photos/artisrams/>)

Old man looking (<http://www.flickr.com/photos/artisrams/3575488218/>)

License: CC BY-ND 2.0 (<http://creativecommons.org/licenses/by-nd/2.0/deed.en>)

Jan Tik (<http://www.flickr.com/people/jantik/>)

Old Couple (<http://www.flickr.com/photos/jantik/4808012/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)

watchsmart (<http://www.flickr.com/people/watchsmart/>)

Old Chinese Men (<http://www.flickr.com/photos/watchsmart/502038199/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)

Jose Manuel Podlech (<http://www.flickr.com/people/jpodlech/>)

Old Aymara woman (<http://www.flickr.com/photos/jpodlech/117588957/>)

License: CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0/deed.en>)

Theresa Knott (<http://www.flickr.com/people/theresaknott/>)

Bulgarian Woman (<http://www.flickr.com/photos/theresaknott/3250186010/>)

License: CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0/deed.en>)

Jaume Meneses (<http://www.flickr.com/people/jaumemeneses/>)

Mallorca'04

(<http://www.flickr.com/photos/jaumemeneses/2092584945/>)

(<http://www.flickr.com/photos/jaumemeneses/2092588307/>)

License: CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0/deed.en>)

Rinaldo Wurglitsch (<http://www.flickr.com/people/wurglitsch/>)

Far de Formentor, Mallorca

(<http://www.flickr.com/photos/wurglitsch/4999221970/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)

Paco Esteban ([http://www.flickr.com/people/paco\\_esteban/](http://www.flickr.com/people/paco_esteban/))

Mallorca\_13 ([http://www.flickr.com/photos/paco\\_esteban/2746772812/](http://www.flickr.com/photos/paco_esteban/2746772812/))

License: CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0/deed.en>)

Maria Rosa Ferre (<http://www.flickr.com/people/mrosa-ferre/>)

La Catedral i el casc de rugby al Parc de mar de Mallorca

(<http://www.flickr.com/photos/mrosa-ferre/5448989582/>)

License: CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0/deed.en>)

daisylaine (<http://www.flickr.com/people/15624270@N03/>)

london theatre (<http://www.flickr.com/photos/15624270@N03/6265353201/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)

Tim Morris (<http://www.flickr.com/people/timmorris/>)

London, England (<http://www.flickr.com/photos/timmorris/3103896345/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)

Andy Roberts (<http://www.flickr.com/people/aroberts/>)

London Bloggers Wine Tasting

(<http://www.flickr.com/photos/aroberts/3133203491/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)

London Theatre Breaks (<http://www.flickr.com/photos/aroberts/4343871012/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)

opopododo (<http://www.flickr.com/people/opopododo/>)

London on the horizon...

(<http://www.flickr.com/photos/opopododo/5038312947/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)

Bruno Cordioli (<http://www.flickr.com/people/br1dotcom/>)

Milano design week 2011 :: Università degli Studi di Milano

(<http://www.flickr.com/photos/br1dotcom/5701445962/>)

License: CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/deed.en>)