

Project ref no	AAL-2013-6-131
Project acronym	Elders-Up!
Project full title	Adaptive system for enabling the elderly collaborative knowledge transference to small companies
Dissemination level	Public
Date of delivery	12/2/2016
Deliverable name	DR2.7 First Functional Requirements and API Specification for Services
Type	Report (R)
Status	Final
WP contributing to the deliverable	WP 2
Main contributors	ISOIN
Other contributors	GEO, IDENER, TUC, CCare
Author(s)	Juan Rodríguez
Keywords	Functional requirements, use cases, API
Abstract (for dissemination)	<p>The purpose of this dissemination level public report is to specify the final functional requirements for Elders-Up! project. The system is comprised of various sub-systems implemented by different partners, which has been interconnected and work together. Requirements are specified both with regards to what functionality the system provides to end users, and with regards to what interfaces and functionality the various sub-systems provide.</p>



**Elders-Up!: Adaptive system for enabling the elderly collaborative
knowledge transference to small companies**

AAL-2013-6-131

Deliverable

**D.2.7 Final Functional Requirements and API specification for
Elders-Up! Services**

Public

© 2014-2017 Elders-Up! consortium

VERSION HISTORY

Version	Edited by	Date	Description
0.1	Juan Rodríguez	15 st September 2016	First version
0.2	Juan Rodríguez	25 September	Second version
0.3	Juan Rodríguez	16 September 2016	Inclusion of input from D4.3, D3.4 and D3.8
0.4	Juan Rodríguez	21 September 2016	Final Draft
0.5	Ionut Anghel	28 September 2016	Review by TUC
0.6	Janna Alberts	1 December 2016	Review by CCare
0.7	Juan Rodríguez	7 December 2016	Final Version

Table of Contents

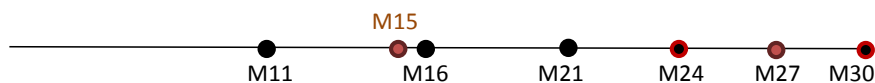
1	INTRODUCTION	6
1.1	GUIDE TO THIS DOCUMENT	7
2	SYSTEM OVERVIEW	9
2.1	USER ROLES	9
2.2	OVERALL ARCHITECTURE	10
3	ELDERS-UP! MODULES AND FUNCTIONAL REQUIREMENTS	11
3.1	INTERFACE SKILL SENIOR (ISS)	11
3.2	DASHBOARD	13
3.3	OPPORTUNITY SELECTION (SMS GUI)	14
3.4	INTERFACE SKILL END USER (ISEU)	15
3.5	ADAPTIVE GROUP SPACE (AGS)	18
3.6	SELF-REPORTING COLLECTION (SRC)	20
3.7	SRC FORMS (SRC GUI)	21
3.8	SENSOR DATA COLLECTION (SDC)	22
3.9	DATA MANAGER (DM)	22
3.10	ADAPTATION DECISION MAKER (ADM)	23
3.11	SKILL MATCHING SERVICE (SMS)	24
3.12	SKILL RECOGNITION (SR)	25
3.13	KNOWLEDGE BASE (KB)	26
3.14	MAILING SYSTEM (MS)	27
3.15	TUTOR	28
3.16	MEDIFORM (FORM MANAGER)	30
4	SYSTEM INTEGRATION	32
4.1	SYSTEM ARCHITECTURE	32
4.2	APPLICATION STRUCTURE	34
4.3	INTERCONNECTION BETWEEN MODULES	34
5	FUNCTIONAL VIEW	35
5.1	FUNCTION SPECIFICATION	35
5.2	FUNCTIONALITY PRIORITY	41
6	USE CASES	44
6.1	USE CASE: CREATE AND CONFIGURE AN ACCOUNT	45

6.2	USE CASE: SIGN IN.....	45
6.3	USE CASE: SIGN OUT.....	46
6.4	USE CASE: UPDATE YOUR PROFILE.....	46
6.5	USE CASE: ACCEPT THE JOB OPPORTUNITY.....	47
6.6	USE CASE: ENTER IN A GROUPSPACE.....	47
6.7	USE CASE: SEARCH FOR A MATCH.....	48
6.8	USE CASE: USE TUTOR.....	48
6.9	USE CASE: CREATE JOB OPPORTUNITY.....	49
6.10	USE CASE: RESPONSE TO A TASK REQUEST.....	49
6.11	USE CASE: SEE SUCCESS CASES.....	50
6.12	USE CASE: CREATE GROUPSPACE.....	50
6.13	USE CASE: SEND MESSAGES OR IMAGES.....	51
6.14	USE CASE: SEND EMAILS.....	51
6.15	USE CASE: MAKE APPOINTMENTS & TRACK APPOINTMENTS.....	52
6.16	USE CASE: ACCEPT/REJECT INVITATIONS.....	52
6.17	USE CASE: CHECK SHARED CALENDAR.....	53
6.18	USE CASE: ADD OR REMOVE FILES.....	53
6.19	USE CASE: EDIT SHARED FILES.....	54
6.20	USE CASE: INVITE NEW MEMBERS.....	54
6.21	USE CASE: ADAPT THE USER INTERFACE MANUALLY.....	55
6.22	USE CASE: ADAPT THE USER INTERFACE AUTOMATICALLY.....	55
7	API SPECIFICATION.....	57
8	LIST OF FIGURES.....	58
9	LIST OF TABLES.....	60

1 Introduction

The Elders-Up! project follows a user-centric system design methodology, in which participatory design techniques are used throughout the project. This deliverable builds upon the foundation of DR2.7 First Functional Requirements and API Specification for Services in which the first functional requirement and system specification was obtained. Following the user driven methodologies that this project embrace, several phases of user evaluation and validation were performed to guarantee an iterative compliance between user needs and the system developed. In particular, the consortium has gone through 2 validations with users before obtaining a final prototype 2.0.

In order to have a clearer view of the milestones (marked in red) and the deliverables associated with WP4 and WP2, we present a brief timeline that comprises the roadmap summarizing the milestones. It should be highlighted that P1.5 (M24) was not included in the initial DoW, however the consortium decided that there were quite a few changes after First Prototype that needed to be shown and tested before the final (second) prototype. As such, it was agreed to develop P1.5 that included all the necessary changes that were taken from the users' feedback of the First Prototype. Table 1 describes milestones and deliverables driven by user feedback and piloting.



Month	Description
M11	DR2.7 First Functional Requirements and API specification for services
M15	D4.2 First Elders-Up! Integrated Prototype
M15	First integrated prototype
M16	D4.5a First prototype evaluation plan
M21*	D4.6b First prototype report (evaluation and recommendations)
M24	D4.5b Final prototype evaluation plan
	P1.5 Prototype1.5

M27	Second integrated prototype
M27	D2.7 Final Functional Requirements and APIs specification for services
M27	D4.3 Final Elders-Up! Integrated Prototype
M30	Final integrated prototype
	D4.7 Final prototype evaluation and users validation

Table 1. Deliverables and milestones from P1.0 to fina prototype

*Moved to M28 for the inclusion of P1.5

In Figure 1 we can see the evolution followed from DR2.7 to D2.7. Deliverable D4.2 describes the first version of the Elders-Up! Integrated prototype where user input and needs have been incorporated through user analysis as described in DR2.7. D4.5b reports on the user evaluation carried out in 3 different pilot sites after the completion of the first integrated prototype. D4.6b presents the recommendation after a deep analysis and understanding on the finding presented in D4.5b. Finally, D4.3 take all this input to deliver a Final Elders-Up! Prototype that fulfils user needs and serve as input for D2.7.

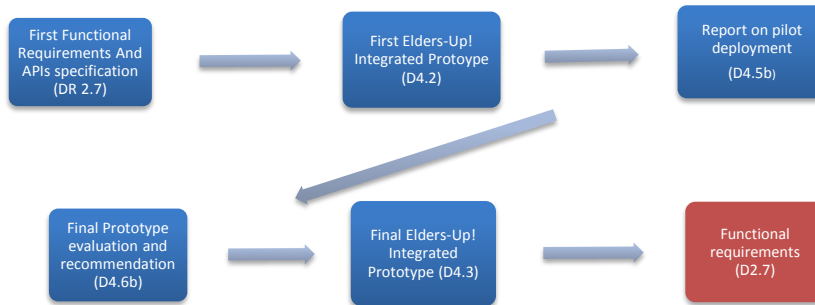


Figure 1. The six stages towards final functional requirements in the Elders-Up user-centric system design methodology

1.1 Guide to this document

This document is a result of the technical development, user specification, user findings and system design work developed throughout the Elders-Up! project. The main goal is to specify

the final requirements; with regards to what functionality the system will provide to end users based on their demands that will be integrated in the final release.

In the following section a brief introduction of the system will be explained. In chapter 3 the functionality of each module of the Elders-Up! system will be described as well as some functional requirement requested by users during the evaluations. Once the functionality of each module is described in chapter 4 the integration between the system modules will be shown. Chapter 5 shows the set of functionalities of the whole Elders-Up! system. Chapter 5 shows an overview of all functionalities implemented with their priority. In order to complete this section the uses cases will be explained in Chapter 6. It is worth mentioning that the API specification that were to be included in chapter 7 has been moved to deliverable D4.3 “Final Elders-Up! Integrated prototype”, for the sake of avoiding unnecessary repetition this information does not appear in this document. Finally, in chapter 8 and chapter 9 the figure list and tables will be shown.

2 System overview

2.1 User Roles

The primary users of the system can be categorized in two groups:

1. Senior Experts (or Older adult User)

They provide experience and knowledge to start-up companies. The aim is to enable this group to share their knowledge and skills, make new connections, meet businesses and volunteer their time through the platform. Each user from this group has some level of expertise in some specific area.

2. Companies (or Company User)

Companies may benefit from the experience and knowledge of the older adults. When looking for a specific skill and knowledge, the company the company can make a new offer in the platform and find the suitable person to be the part of the company's team.

In the user research carried out prior to the first prototype (Refer to DR2.7), an optional third role was identified:

Moderators

The moderators facilitate the matchmaking process. They can for example support companies and experts in creating their profiles, in finding matches, and in starting a collaboration process.

This role has been played out by the evaluators during the various pilots and validation phases in order to ease the interaction of users and companies.

After the completion of Prototype P1.0 and based on the functional requirement explained in the next section, the consortium opted for the development of an online virtual tutor (For a detail description of this functionality refer to D3.4).

2.2 Overall Architecture

The Elders-Up! system is composed of several modules that can be seen in Figure 2. This architecture has been improved through changes with respect to the architecture presented in DR2.7. These are the inclusion of the MediForm Form manager and the Tutor. User requirement leading to this change will be covered in chapter 3.

For a complete system description including all the modules refer to:

- D4.3 Final Elders-Up! Integrated prototype: Complete system architecture and APIs.
- D3.8 Collaborative and adaptive workspace 2nd prototype: ICAW including interfaces and final functionalities of the platform.
- D3.2 Skill Matching service 2nd Prototype.
- D3.4 Sensors and self-reporting data gathering 2nd Prototype
- D3.6 Adaptation Decision maker 2nd Prototype

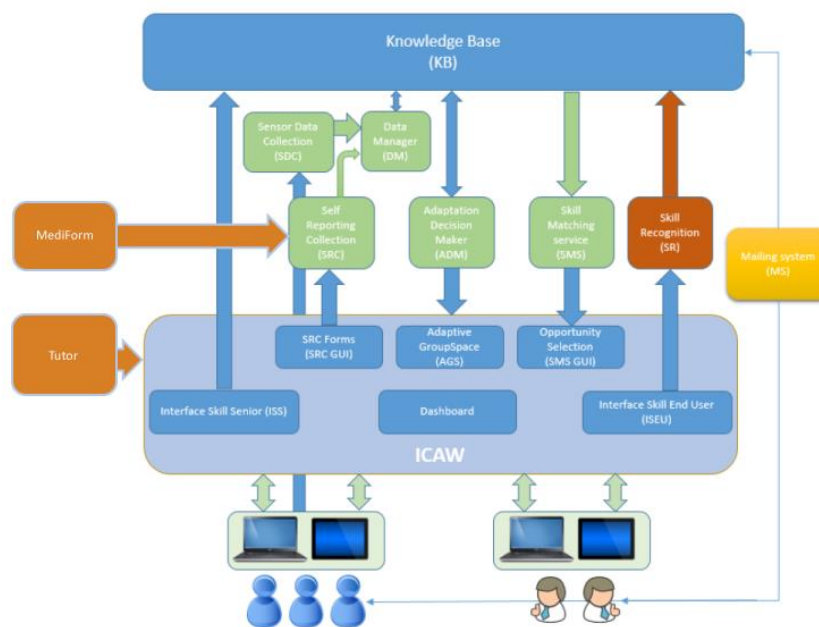


Figure 2: Elders-Up! application structure

3 ELDERS-UP! Modules and Functional Requirements

In this chapter the functional requirements of the different modules will be described. In addition, this section describes the features achieved by the final prototype for each one of the modules located in the internal architecture shown in the previous section.

3.1 Interface Skill Senior (ISS)

The Interface Skill Senior (ISS) is the GUI (Figure 3Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.) that enables the senior to:

- Set up/amend profile
- Include availability options for collaborations
- Express motivation to collaborate
- View current opportunity matches
- Accept/reject opportunity matches

The ISS allow the senior to input all of their profile information. It stores all of the relevant personal information, together with the users' skills inputted either by the skills taxonomy (Figure 4 and Figure 5).

Users are able to amend profile settings and skills.



The screenshot displays the 'Profile' page of the Elders-Up! application. The page is titled 'Profile' and has a blue header bar with a 'Back to dashboard' link on the left and a 'Help' icon on the right. The main content area is divided into three tabs: '1. Personal Information', 'Employment and Skills', and '3. Motivation'. The 'Personal Information' tab is active, showing a profile picture of Jose Antonio Carvajal and a list of personal details. The details include: Name (Jose Antonio Carvajal), City (Seville), Country (Spain), E-mail (carvajal.isoin@gmail.com), Phone number (991888555), Mobile number (691888555), Skype name (skype), Preferred contact method (Email), and Language (English). There are 'Save and continue' buttons next to the 'E-mail' and 'Language' fields, and a 'Change Password' button under the 'Account Setting' section.

Figure 3. Personal data

Profile

1. Personal Information 2. Employment and Skills 3. Motivation Save and continue >

Previous employment position
Senior Web Technologies Consultant

Languages spoken *
×English ×Spanish ×Italian

Competences *
Click in the box below and select 2 or 3 competences that describe what type of person you are.
×Ambitious ×Amiable ×Analytical

Skills *
Please add your skills below. Around 2-3 skills would be enough to allow companies to find your profile although you will get more opportunities when including more skills. You can add new skills by using the search function below or by clicking the add buttons on the right hand side of skills tree. To remove one, just click the trash icon at the left of the skill name.

Your skills
Computing IT Developer Support

Figure 4. Senior user skills

1. Personal Information 2. Employment and Skills 3. Motivation Save and continue >

Motivation
What would you like to share about yourself, for example what motivates you to join this platform

I really want to help others to go forward at a faster pace thanks to my gathered knowledge gained through my professional career. I really like working with people.

Availability
Let the EldersUp!'s companies know when would you be available

Full time availability, preference for mornings

Cancel Save and continue >

Figure 5. Motivatoin and availability

The main findings found during user evaluations of this module are shown in Table 2:

Register and profile completion
All registration fields shown were set as obligatory which was perceived as too much information to fill in at once.
The language drop down menu was not ordered in an alphabetical order, making it confusing for users to find the correct language.
If the user went back to the previous page, the information was not automatically saved
Users did not understand the aim of the profile page, and needed a short introduction.
The skills tree was perceived as complex and unclear.

Table 2. End user Requirements

These findings have been analyzed and addressed in the following way: (1) Introducing the profile page during the first time visit using introduction modals, (2) Languages are ordered alphabetically, (3) Information is automatically saved and (4) improvements in the usability of the skills tree

3.2 Dashboard

The dashboard is the central menu page for the senior experts and the SME's (Figure 6). It is worth noting that the Dashboard shows a slight different functionality whether the user is a senior or a company. The dashboard provides access to the Adaptive Group Spaces, to the user profile, company profile, and to the search & match functionality. Additionally, if the user is a company it will have the option to create a GroupSpace for other seniors to join. Functional requirements can be summarized in:

- Easy access to Elders-Up! core functionalities

The look and feel have been greatly improved based on user feedback. The current system is focused on usability and understandability. A clean design and icons representing the features underneath shows this approach.

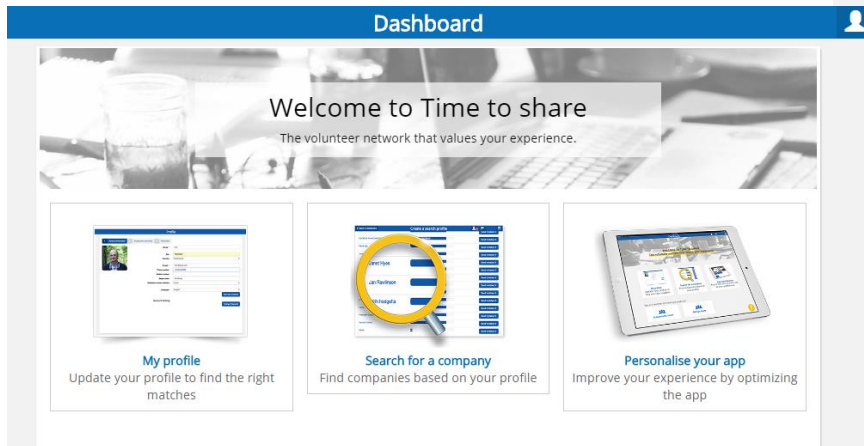


Figure 6. Dashboard senior expert

3.3 Opportunity Selection (SMS GUI)

The Opportunity Selection (SMS GUI) is the interface from where users select matches that are provided by the skills matching service.

The SME matching selector provides a list of senior users matched to the job opportunity input through the skill recognition (SR) module and is output via the SMS. The matches are weighted according to the algorithms within the SMS and ranked accordingly. The SME can then decide which of these matching profiles they wish to work with. At the same time the seniors are also able to decide which of the matching companies they wish to work with. The SME then contacts the user and invites them into the ICAW to collaborate.

Within the senior user matching selector, the senior is provided with 2 options:

- The chance to accept (or reject) a collaboration request from an SME (as detailed above). Should the user accept, then they are invited to collaborate in the ICAW along with the SME. Should they reject then the SME is notified and can then choose another potential match.
- The chance to browse current opportunities that the system calculates that the user has a high matching score with. The user can then contact

the SME and request a match which, if accepted, they can both collaborate.

The user can also view current matches and have the opportunity to accept/reject any jobs they have been matched with. As Figure 7 shows the current step of the collaboration process is displayed in a very graphical way to improve user readability.

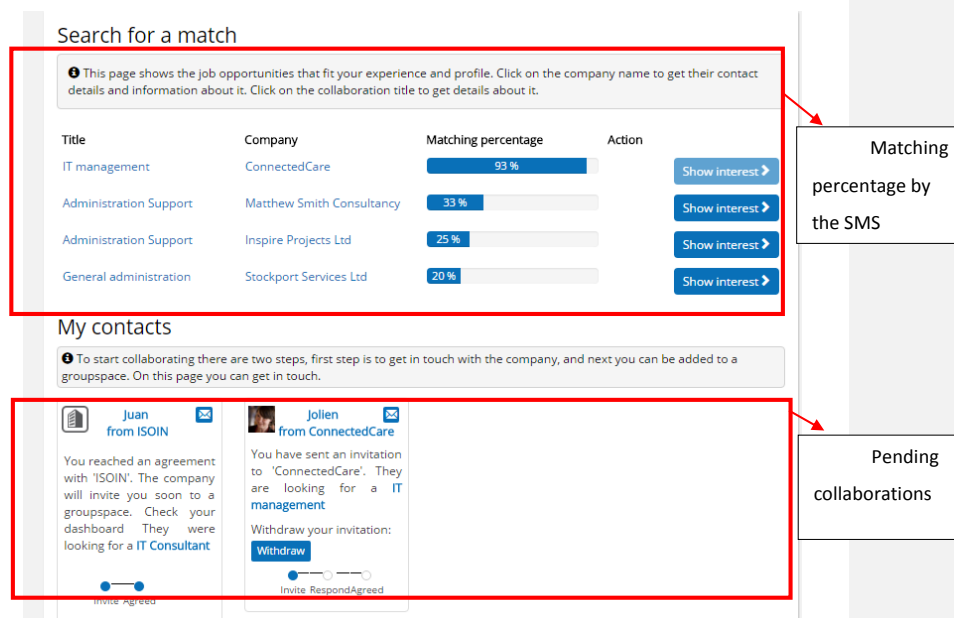


Figure 7. SMS GUI

3.4 Interface Skill End User (ISEU)

The interface Skill End User (ISEU) is the GUI that enables companies to:

- Set up/amend profile
- Add job opportunities
- View current matches
- Select users and invite them to collaborate.

The ISEU allows the company to input a company profile which will be stored in the Knowledge Base (KB). It also allows the user to input a job opportunity and include a list of required skills, which will then be matched by the SMS to give a list of

matching users. This module is really similar to the ISS previously presented for the senior users.

The main flows encountered during the evaluation leading to the final prototype design are extracted from D4.6b and summarized in Table 3:

Registration
The process of finding a match could have been a bit faster without too much details to fill in during the registration
The registration process seemed a bit long, and they were not sure which information would be shared. Not all fields should be obligatory
Flow for finding a match
Flow of finding a match could be improved
The matching details could be clarified
The skills tree could be optimized; it was not directly clear
Companies would like to see directly which seniors were using the platform
The flow of the application seemed a bit complicated and could be improved

Table 3. End User Requirements (2)

All requirements presented in **Error! Reference source not found.**3 have been addressed as new functionality. The following figures (Figure 8, 9 and 10) show sequentially: (1) Reduced steps needed for registration process (2) Available seniors of the platform during the matching process (3) Search engine easing the matching process

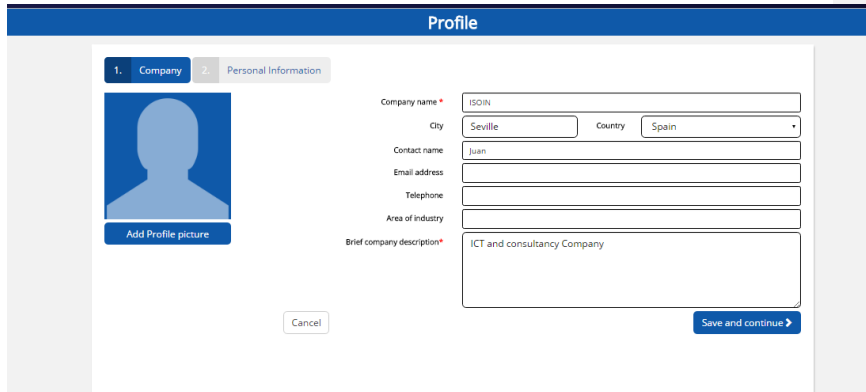


Figure 8. ISEU GUI with simplified obligatory fields

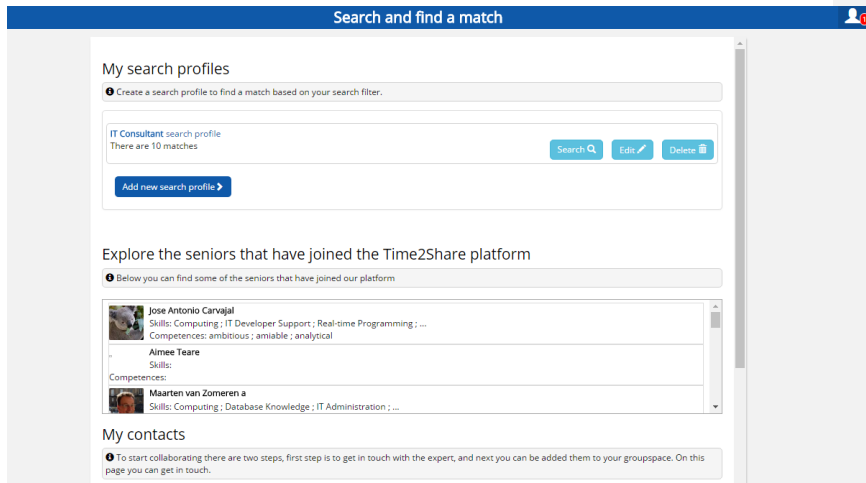
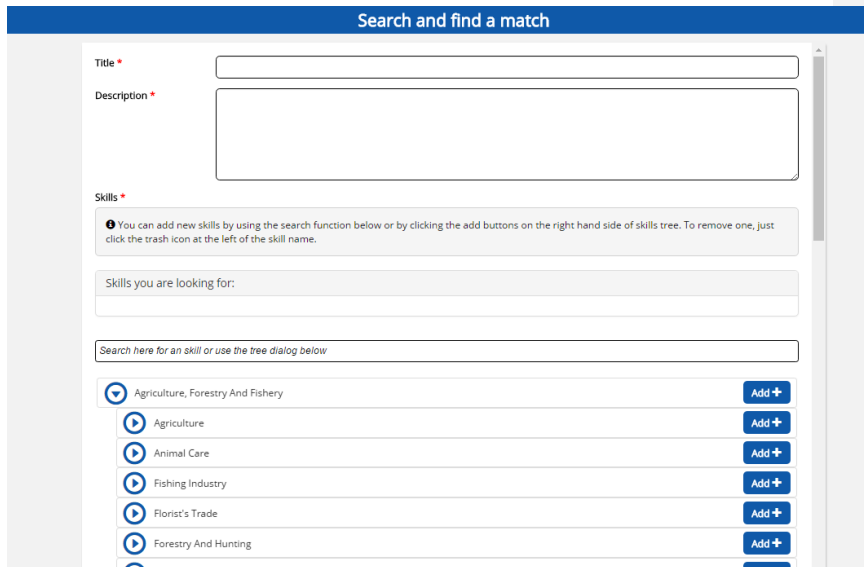


Figure 9. Search GUI with available users



The screenshot displays a web interface for creating an opportunity. At the top, a blue header reads "Search and find a match". Below this, there are three main sections: "Title" with a text input field, "Description" with a larger text area, and "Skills". The "Skills" section includes a help message: "You can add new skills by using the search function below or by clicking the add buttons on the right hand side of skills tree. To remove one, just click the trash icon at the left of the skill name." Below the help message is a search input field labeled "Skills you are looking for:". Underneath that is a search bar with the placeholder text "Search here for an skill or use the tree dialog below". A list of skills is shown below the search bar, each with a circular icon on the left and an "Add +" button on the right. The skills listed are: "Agriculture, Forestry And Fishery", "Agriculture", "Animal Care", "Fishing Industry", "Florist's Trade", and "Forestry And Hunting".

Figure 10. Opportunity creation GUI

3.5 Adaptive Group Space (AGS)

The GroupSpace is the central location to support companies and their teams of experts in their day-to-day collaboration. Companies can ask for support, and both company members and experts are facilitated in communication, coordination and compensation. Functional requirements include:

- Message system to interact with other GroupSpace members
- Calendar for appointment management
- File sharing
- Task management system allowing for an easy task assignment

The workspace consists of different elements (**Error! Reference source not found. 11** and **Error! Reference source not found. 12**). Requests can be used for task management. Within the collaborative agenda, the team can manage and share their appointments. Contacting each other is made easy with the group messages. The messages are sent to the entire group. It is also possible to share documents.

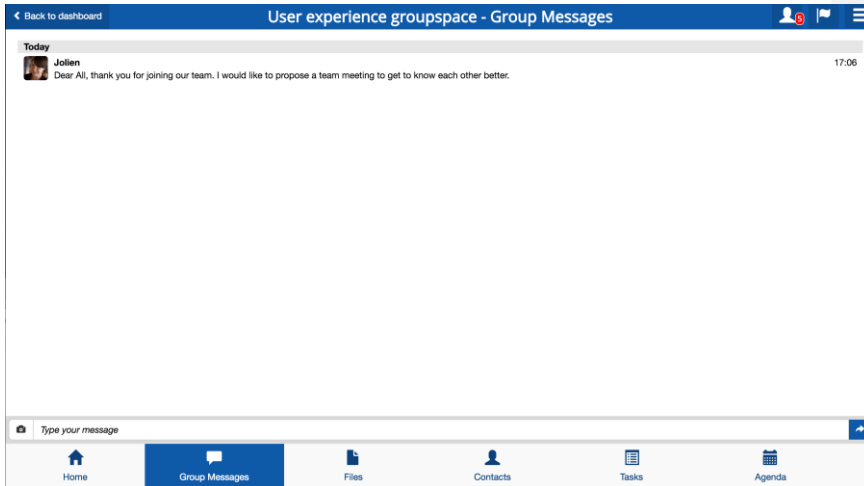


Figure 11. Group Messages

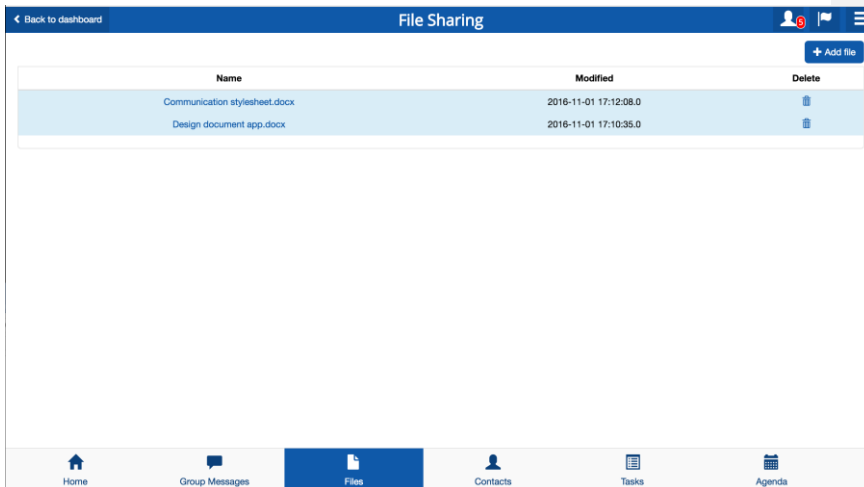


Figure 12. Files sharing system

The Adaptive Group Space can adapt itself to the cognitive conditions or physical limitations. The Elders-Up! system addresses these varying user capabilities by offering adaptation.

More details and requirements can be found in D3.8 Collaborative and adaptive workspace 2nd Prototype.

3.6 Self-Reporting Collection (SRC)

The main goal of the Self-Reporting Collection (SRC) is to manage the different questions that will be asked to the End-user. The general architecture designed for this module can be seen in Figure 13.

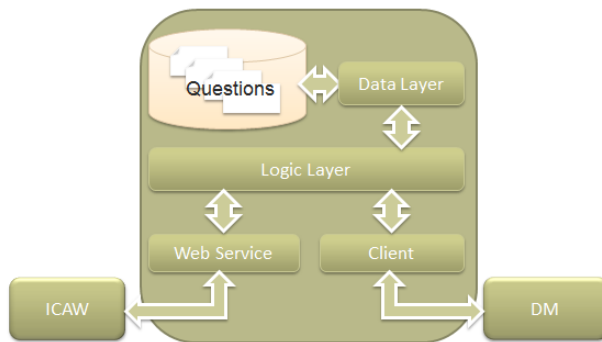


Figure 13: Self-Reporting Collection structure

As we can observe the SRC module communicates with the ICAW, which generates the necessary interfaces for the SRC and with the Data Manager (DM) to output detected problems with the user.

Two modules are in charge of managing communications, Web Service and Client:

- **Web Service:** This module provides ICAW with services relative to questions supply that will be asked to the user besides it will feature a service for gathering the responses from users.
- **Client:** This module is in charge of making requests to another Web service from the Data Manager to send parameters that are used to identify an impairment from senior users though the specific problem is not decided here.

The most important module within the SRC is the *Logic Layer* which has the application logic and it is the part of the system that decides which type of questions should show to the user.

Questions are stored in an internal data base and the logic layer responsible for managing these questionnaires inside the data base is the *Data Layer*.

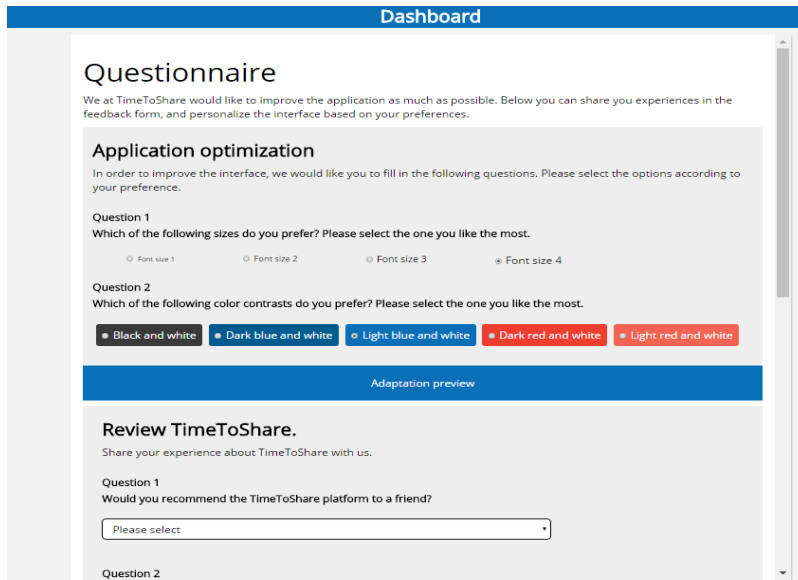
For a complete description of the module please refer to D3.3 and D3.4.

3.7 SRC Forms (SRC GUI)

SRC Forms is the section of the ICAW that handles the graphical user interface from the SRC. Different forms composed in the SRC stages are visualized through these custom interfaces generated dynamically for each user based on its profile. The main characteristics of the interfaces composed of several forms are determined aiming at the user convenience. The main functional requirements of the interface adaptation process are:

- Brief questions with no dense elaboration.
- Question address in a simple way the possibilities of adaptation regarding letter size and background contrast

Several improvements have been made to address this previous points as shown in Figure 14 below.



The screenshot displays a web interface titled "Dashboard" with a "Questionnaire" section. The questionnaire is divided into two main parts: "Application optimization" and "Review TimeToShare".

Application optimization
In order to improve the interface, we would like you to fill in the following questions. Please select the options according to your preference.

Question 1
Which of the following sizes do you prefer? Please select the one you like the most.

Font size 1 Font size 2 Font size 3 Font size 4

Question 2
Which of the following color contrasts do you prefer? Please select the one you like the most.

Black and white Dark blue and white Light blue and white Dark red and white Light red and white

Adaptation preview

Review TimeToShare.
Share your experience about TimeToShare with us.

Question 1
Would you recommend the TimeToShare platform to a friend?

Question 2

Figure 14. SRC GUI

3.8 Sensor Data Collection (SDC)

The main role of the module Sensor Data Collection is to collect the information coming from the different sensors and perform the pre-analysis of the data obtained, the goal is to provide the user with an smart system able to adapt automatically according to the user interaction. This module is based on three main blocks. For a complete and detailed description of this module please refer to D3.3 and D3.4.

3.9 Data Manager (DM)

Data Manager is the module where data coming from SRC and SDC is merged and processed. In Figure 15 we can see the general architecture of the Data Manager module. Data manager functional requirement have not suffered any changes from P1.0 onwards.

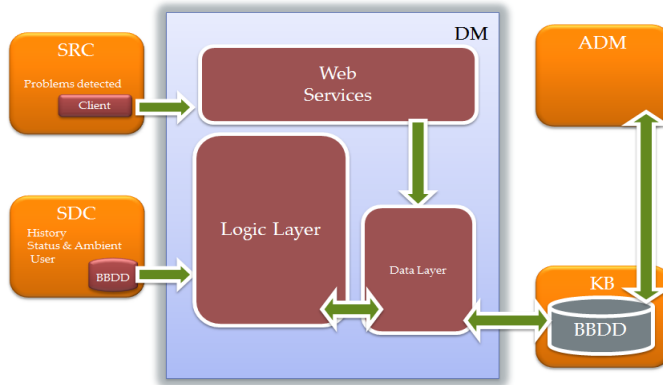


Figure 15: Elders-Up! DM System

3.10 Adaptation Decision Maker (ADM)

The Adaptation Decision-Maker (ADM) decides and selects the best configuration of the Interface Collaborative Adaptive Workspace (ICAW) to visually display the information to end users while fitting their preferences and helping them to overcome limitations (e.g. visual impairments or cognitive limitations) and empowering their engagement in working collaboratively with others through the Elders-Up! platform.

Updates integrated in the final prototype are:

- Move from standalone adaptation decision making service to integrated component of the Elders-Up! Platform (within the web browser)
- Algorithm improvement and adaptation features prioritization
- Updated Knowledge Base data model and integrated the data access layer component
- Added a decision tree algorithm for style generation
- Adjustments to the data and interaction flow
- Adjusted data and interaction flow

The final ADM design architecture is shown in Figure 16. Additionally, Figure 17 shows an example of an adaptation performed thanks to the module decision making.

For a complete description of the ADM and the internal functioning please refer to D3.6.

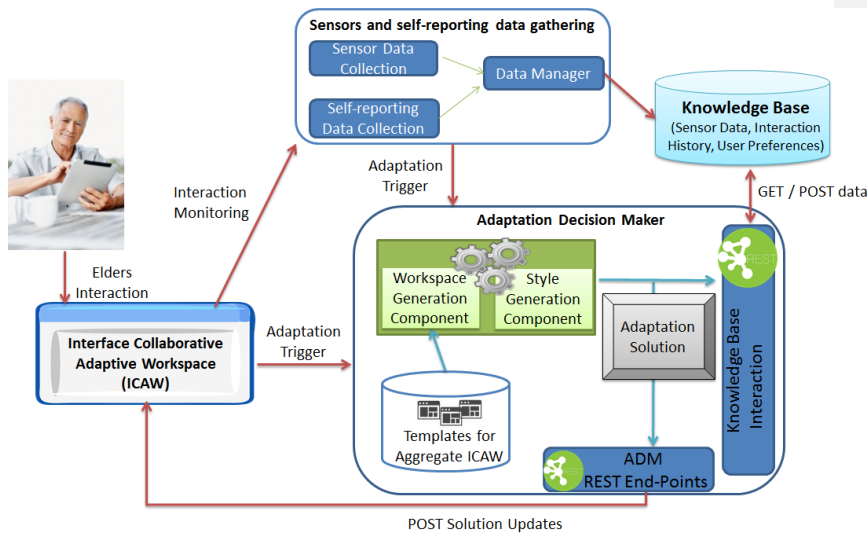


Figure 16. Adaptation Decision-Maker design

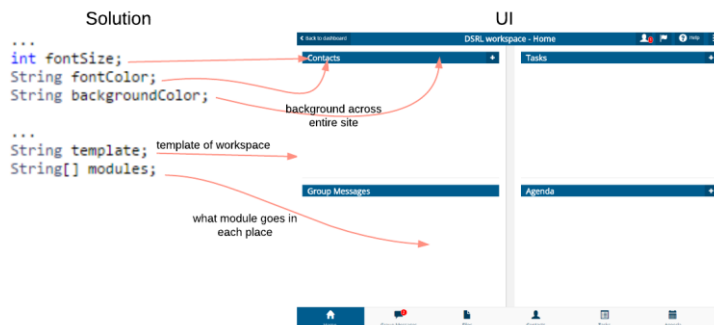


Figure 17. Example of adaptation solution and associated ICAW UI

3.11 Skill Matching Service (SMS)

The main objective of the Skills Matching Service (SMS) is to cross-compare the skills offered by the elderly end users and those required by the small companies and start-ups with the goal of finding an optimal match. Figure 18 shows the logical architecture and relation to other components.

Updated features implemented into the second prototype are:

- Implemented a more efficient version of the 1-to-1 Skills Matching algorithm
- Implemented a new algorithm for constructing the multidisciplinary workforce of seniors for a job offer (N-to-1 Skills Matching).
- Implemented the skill discovery from free text functionality, this has been implemented using text pre-processing and multiword predefined patterns for new skills identification.
- The Added language clustering for both 1-to-1 Skills Matching and N-to-1 Skills Matching
- Added a dynamic threshold for the results provided by 1-to-1 Skills Matching

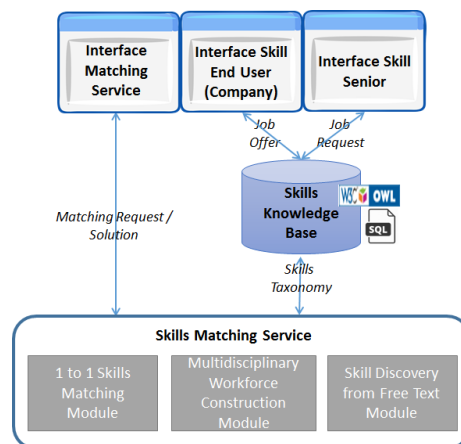


Figure 18. Final module logical architecture and relation to other Elders-Up! Components

A full description of the module, architecture and algorithms is described in D3.2.

3.12 Skill Recognition (SR)

The skill recognition module (SR) is the interface which allows the SME to input the opportunities into the KB along with the skills required and can be seen in Figure 19.

The SME enters the required skills via the taxonomy system similar to that which has been developed for elderly users to enter their skills. The tree system means that branches as well as leaf nodes can be selected. This means that the SME can be exact

in their skill requirements or more general. The branch would then include a subset of skills which are then ranked and matched within the SMS accordingly. (i.e. exact skill matches are ranked higher than subset skill matches).

This job opportunity, once entered is then stored in the KB and is also processed in the SMS to feedback matches to the SME.

Search here for a skill or use the tree dialog below	
▼ Agriculture, Forestry And Fishery	Add +
▶ Agriculture	Add +
▶ Animal Care	Add +
▶ Fishing Industry	Add +
▶ Florist's Trade	Add +
▶ Forestry And Hunting	Add +
▶ Horticulture	Add +
▶ Viticulture	Add +
★ Nature Protection And Landscape Conservation	Add +
▶ Architecture And Building	Add +
▶ Arts	Add +
▶ Business And Administration	Add +
▶ Computing	Add +
▶ Education	Add +
▶ Electrical Engineering	Add +
▶ Environmental Protection	Add +
▶ Health	Add +
▶ Humanities	Add +

Figure 19. SME GUI for entering skills requested for a job opportunity

3.13 Knowledge Base (KB)

The knowledge base is the database that stores all of the user profile information, the taxonomy of skills, job opportunities and sensor data collected by the SDC. It consists of four primary data tables, storing data specific to users, along with several ancillary tables storing data used by the application itself (e.g. the skills taxonomy and list of languages) and some pivot tables that enable the use of many-to-many relationships.

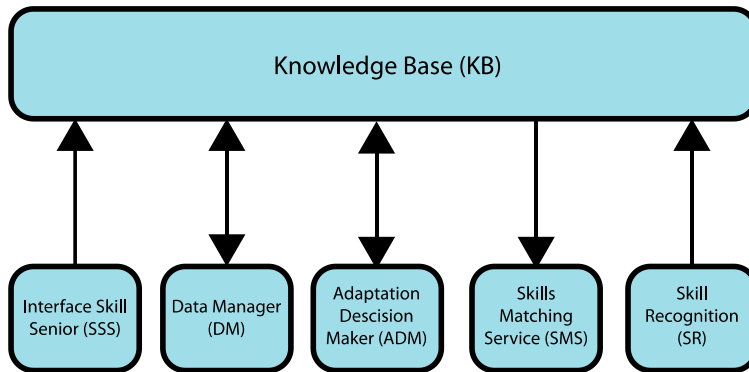


Figure 20. Knowledge base logical connection to other components with in/out flow of the information direction between modules.

Figure 20 shows the principal modules that interact with the KB. Namely

- Interface Skill Senior (ISS)
- Data Manager (DM)
- Adaptation Decision Maker (ADM)
- Skills Matching Service (SMS)
- Skills Recognition (SR)

Data flows from other modules into these and then in and out of the KB. For a detailed description of the whole system please refer to D3.5.

3.14 Mailing System (MS)

The mailing system takes care of the communication of the system to the users, senior experts and company users. It is meant to motivate and involve users in the Elders-Up platform. In some cases, it is aimed at inviting users to respond to an invitation: e.g. invite for the following group space or appointments. In other cases it is to update users of what is happening in the group space: e.g. these requests have been performed and these messages have been send (Figure 21 and Figure 22).

Technically the mailing system works in two ways: either send invitations or notifications are done upon user input. So the mailing system exposes a few functions that can be triggered by other modules. Informing about the status of the system is triggered from within the Mailing System. It sends this information when a remarkable event occurs, e.g.: a registration or collaboration match.

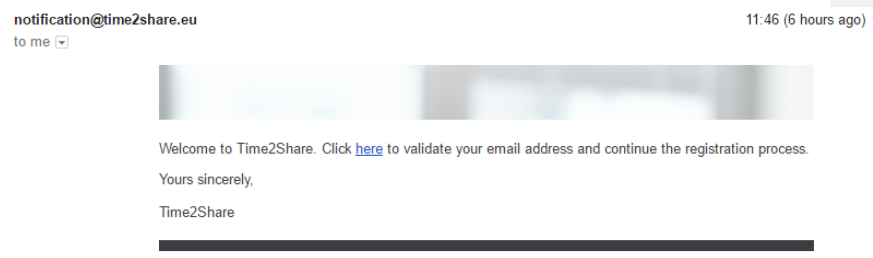


Figure 21. Email notification sent by the platform

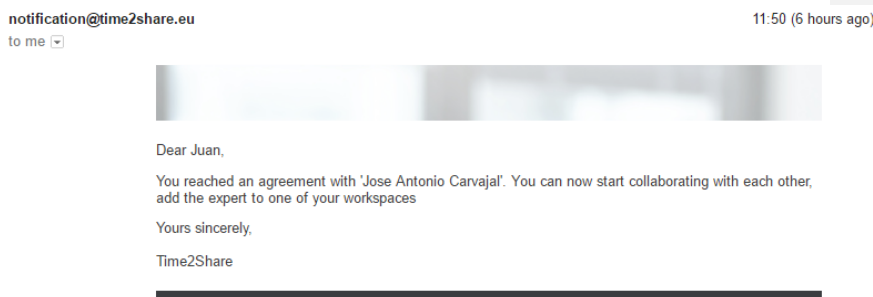


Figure 22. Email notification sent by the platform (2)

3.15 Tutor

The tutor is the module in charge of delivering assistance both on user request and in case assistance is deemed necessary. Figure 23 shows the button to receive help on demand.



Figure 23. Tutor help button

The tutor helps users to do any task that can be performed within the Elders-Up! In two different ways; by an easy “step by step” based in short descriptions and item

highlighting and by videos of the task recorded by an expert user. For that purpose, it needs to interact directly with the front-end, which is why a JavaScript based model-view-controller was designed in order to be able to show instructions, record times, highlight elements, and check interactions. The system architecture of the tutor is shown in Figure 24.

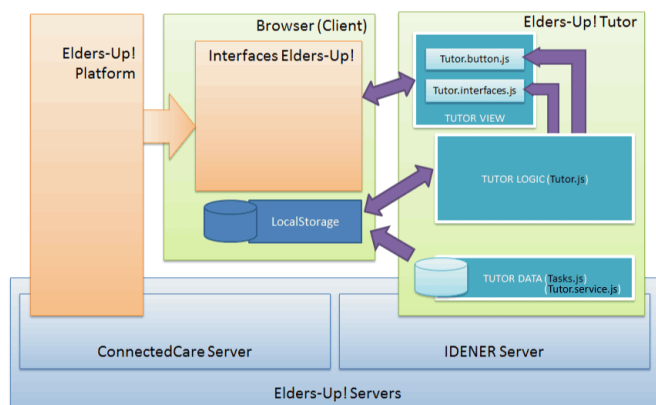


Figure 24. General architecture of the tutor module

The main menu GUI is shown in Figure 25, from that menu the user is able to select the different sub-actions inside the platform. Help provided to the users is also context sensitive, meaning that the functionalities and help shown are based on the flow of the user is in. For a complete description of the module along with obtained functionalities and description of the internal algorithms and architecture please refer to D3.4.

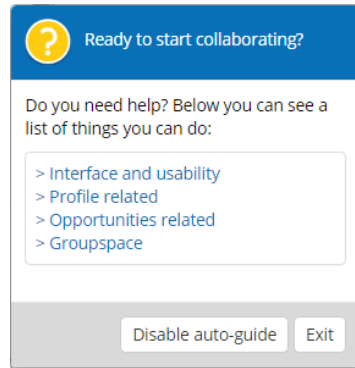


Figure 25. Tutor main menu

3.16 MediForm (Form Manager)

MediForm is a separate module which allows the end user to enter personal information on their subjective wellbeing and their perception on the platform. The module was created based on the discomfort from some users to answers questions about their possible disabilities or impairments inside the general questionnaires delivered by the SRC. Using the MediForm module, the platform emphasizes that the questions are voluntary, anonymous, and not required to Elders-Up!

The MediForm module is an external tool which can be used to collect data through questionnaires and potentially through interactive games, and has been developed to support the user evaluations (Figure 26 shows the main page). The module will not be part of the Elders-Up! platform when it is used in a commercial setting. The module can generate statistics regarding for example usability, user's behavior, and engagement with the platform.



Figure 26. MediForm main page

MediForm is built using a MVC architecture with Spring, using Struts2 to create dynamic forms. MediForm is a completely independent module, allocated out of the Elders-Up! platform, what means that it acts transparently to the platform. Figure 27 shows a class diagram of the solution.

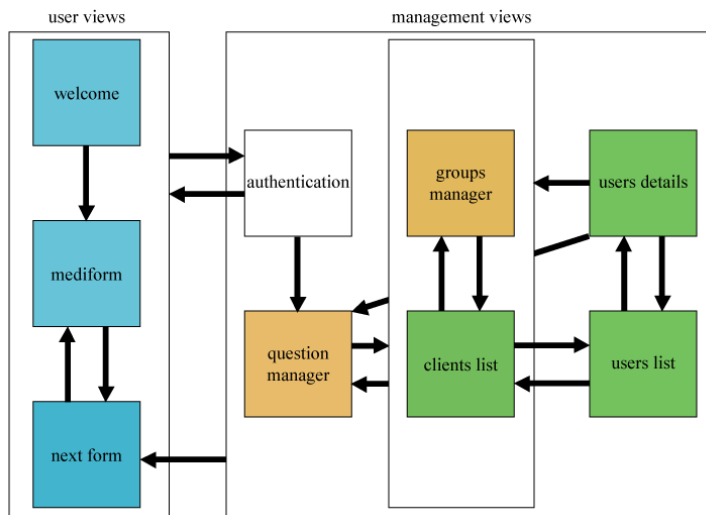


Figure 27. MediForm platform schema

For a complete description of the system refer to D3.4.

4 SYSTEM INTEGRATION

In this section the final system architecture designed and implemented for the Elders-Up! Prototype is depicted. For the development of the platform a MoSCoW technique have been selected to classify the most important features, thus the characteristics have been classified depending on its priority and has been implemented sequentially throughout the project.

4.1 System Architecture

The Elders-Up! Platform has moved from a centralized server architecture (foreseen in the platform mock-up) to a distributed architecture (Figure 28) in order to ensure the protection of IP for the developed modules and to allow a better load management. For the final prototype two main servers have been prepared for the platform and an additional one to host the home page of the platform.

The first one (Located on Idener network) and referred as EUP-SERVER-1 from now on, stores the different services developed by the consortium, the main system database EUP-DB-1, and the views associated with the services and profiles of the users. EUP-SERVER-2 (Located on CCare network) operates as a gateway to access the EldersUp! Platform, offering the registering services, serving the HTML pages and including all the functionalities integrated in the *GroupSpaces*. The following figure depicts the changes in the architecture from the first concept to the current implementation:



Figure 28.Final Elders-Up! global system architecture

The application has been developed as a full compatible HTML5 dynamic web page applying a responsive design. This guarantees usability and compatibility with a wide range of devices (computers, mobile phones/tablets, etc.).

Regarding security, personal information of the user is treated confidentially and several security measures have been integrated in order to satisfy user expectations. In the first prototype, the communication of the users with the platform is made through an un-encrypted http regular connection. Both EUP-SERVER-1 and EUP-SERVER-2 have been set-up taking into account security concerns in order to assure personal data protection. For final prototype, the communication between the server and the devices has been encapsulated in a secure session, using https protocol. Further improvements on this area have also been developed as database encryption techniques and SQL injection prevention.

To host all the services that are to be provided, the following detailed technical infrastructure is being used:

EUP-SERVER-1: EUP-SERVER-1 is located in the IDENER cloud platform. It runs as a virtualized server using PROXMOX technology over a quad cored XEON based system with 32GB of RAM with 2x2 TB Hard disks in RAID configuration. The operating system is Ubuntu LTS 14.04 64 bit. User accounts for all the developers have been established and the required software packages have been installed. Specifically, Apache 2 is used for serving the web-pages with PHP module enabled and PHP 5 installed. MySQL database server is used for DB handling. Tomcat 8 has been deployed to support JAVA services.

EUP-SERVER-2: EUP-SERVER-2 is within the cloud platform of Movements Group. It is a virtualized server running Microsoft Windows Web Server 2008 R2 and has 4 GB of RAM. All the user accounts for the developers are established. The virtual machine runs JAVA 8, Tomcat 7 and MySQL 5.6. Backups are done on a daily basis. The server runs apache tomcat and MySQL.

For a complete and detail description of the system architecture and the services delivered on both servers refer to D4.3.

4.2 Application Structure

In Figure 2 a scheme of the Elders-Up! Application architecture is presented. The list of the different modules is presented in the previous chapter 3 as well as the different inputs and outputs. In the next paragraphs some details regarding how these modules are being integrated is provided.

4.3 Interconnection between modules

The information exchanged with the users is performed through a series of interfaces designed to fulfil user requirements. Therefore, two different GUI (Graphic User Interfaces) have been developed, one for each of the target groups of the application.

In order to properly process the users' information, different services have been implemented: the Skill Matching Service (SMS), the Sensor Data Collector (SDC), the Adaptation Decision Maker (ADM) and the Adaptive GroupSpace (AGS). New modules added in the system architecture are the MediForm online questionnaire and the Tutor. More information regarding these services is provided below in the corresponding sub-sections. Additionally, the application handles the offline (i.e. email) communication with the user through the Mailing System (MS) which has been developed for the final prototype.

All the dynamic contents (HTML pages generated through PHP) are served through the EUP-SERVER-2 which internally connects to EUP-SERVER-1 to retrieve the required contents. In the user side, users connecting to the platform through a PC, will be accessing the HTML5 version of the platform, and therefore will not need anything special besides an HTML5-compatible web browser.

5 Functional View

This chapter lists system functionalities added in the final Elders-Up! platform. These functionalities are presented from an end user perspective; what the user can or cannot perform in the platform.

5.1 Function Specification

The functionalities are grouped in main categories, designated by letters, with a numbered list of functions in each category, so that we can refer to functions in this form: A.1, D.4, etc.

A. Manage user account

In order for a primary user to have an account, a profile needs to be created.

A.1. User Registration

An account is created through the Elders-Up! system using an email address and password for authentication. All end users can enter in their contact details:

A.1.1 First Step: Add personal information.

The first step to registration is for users to fill in their personal information such as: their first and last name and their email address. Address and contact phone are optional.

A.1.2 Second Step: Add Employment and Skills.

The second step allows the older adults to add their employments and skills using a structure of the simple selection (tree based) with all the possibilities which allow the selection in less than three clicks. Additionally, it is possible to add skills through a search field incorporating an autocomplete functionality. Both selection methods are designed in a way to make the selection of skills easy and fast for older adult users.

A.1.3 Third Step: Tell us about yourself.

The third step allows the user to put the other information such as motivation and availability. This free text field will be processed from the user input so the important information for the future matching can be recognized.

A.2. Update your profile

End users can modify any field of the user profile. User can update for example the address or add a new skill and save it again.

A.3. Sign In

The first action to enter in the platform is to login, so the users can use the system.

A.4. Sign out

If the user wants to leave the Elders-Up! system he/she can sign out. When the user goes out of the platform the previous temporary private information of the user is deleted, so he/she needs to sign in if wants to use the platform again.

B. Primary Actions

Primary actions are those which are not related to any specific job, and can be accessed from the main menu.

After the login the first window for the user is a Dashboard where end users (older adults and companies) can choose between the different primary actions in the platform.

B.1 Show Job opportunities

Users can see the list of suggested job opportunities based on a personalized match performed by the system. Each of the opportunities has the following characteristics:

- Percentage of the match between the older adult skills and job opportunity
- Show company's information and job opportunity
- Skills required for the job

- The responsible of the opportunity and his/her profile
- Link for chatting with the responsible of job opportunity
- Link to accept the opportunity
- Description and another features of the job opportunity

B.2 Create a GroupSpace

Companies can add in seniors that have accepted a collaboration and create a new workspace where this previously contacted senior can collaborate with them.

B.2.1 View job opportunities

Companies can see the different job opportunities generated and edit or delete the existing ones.

B.2.2 Edit job opportunities

Companies looking at job opportunities can press the edit button and a form appears where company can change the details and save it again.

B.2.3 Delete job opportunities

Companies can delete the job opportunity, removing the job opportunity from the system will update the current list removing the opportunity deleted so it is not shown any more to the older adult user.

B.3. Enter to GroupSpace

In the dashboard the user has the option to enter a GroupSpace previously created by the company which they have been invited to.

This GroupSpace will provide the user with the necessary tools to collaborate with the companies and other seniors added in by the company.

B.4 Search for companies or experts

The users can use the search engine to look for companies or senior experts. The search option is available from the dashboard or main page of the Elders-Up! system. If the user is older adult the option will be a company search, for companies the option will be senior search.

B.5. Tutor

The tutor is located in the upper right corner of the dashboard. This module offers help on demand on all the tasks contained in the platform. Additionally, the tutor will provide help if it detects that the user is lost or disoriented in the platform. The help is offered via videos or step by step interactive guide.

B.6. Use Request module

The older adult experts can receive collaboration requests from companies requiring their skills. This module is in charge of the communication between the end users (older adults and companies). This module is consists of:

B.6.1 Add tasks requests

Create new request and fill in the next values:

- Deadline of the request
- The type of the request (a question, an office task or coaching)
- Assign the request to a team member

B.6.2 Respond to a task request

User can respond to the requests by accepting the invitation or declining the invitation.

B.7. Success case module

This module shows the successful matches. All the users can enter and see these matches.

C. GroupSpaces

GroupSpace is the environment inside of the platform where the users are interacting between themselves regarding the specific job opportunity.

During the first time visit introduction modals explain each functionality to the end users.

Inside of each GroupSpace the user can see the different objects and attributes:

- Members group and their contact details

- Group messages
- See the To-do-Tasks and see status of task process
- Documentation repository
- Calendar

C.1 Manage GroupSpaces

A company administrator is be able to manage the settings for a workspace, e.g. manage who has access to the workspace, and what information is available to them.

C.2. Send messages

Share pictures and text with other members of the GroupSpace in the platform.

C.3. Send email

The platform can show a link to the external e-mail client with the recipient to all members of the GroupSpace. This way users within the GroupSpace may email the other members in an easy manner.

C.4. Start voice communication

All users can start video conversations with other members using the GroupSpace that is linked to the skype.

C.5. Start video Communication

This option has been discarded

C.6. Make and track appointments

All users can create and follow up created appointments. Actions available within this module are:

C.6.1. Create appointments (one-time and recurring events)

To create an appointment the user needs to fill in the next fields:

- Set date and time for the appointment
- Select a team member
- Join an appointment

C.6.2 Accept invitations

All older adult experts can accept the invitations sent by companies.

C.7. Show a calendar common in the GroupSpace

All appointments appear in the calendar that is accessible for all the users that are members of the specific group (with the same job opportunity) of the workspace.

C.8 Use file sharing module

All users can share files inside of this module available for each GroupSpace. Allowed operations are:

C.8.1 Add or remove Files

Users can add new or remove existing files in the shared repository.

C.8.2. add or remove folders

This option has been discarded on the final prototype.

C.8.3. Edit shared files

Users can edit existing files on the platform repository.

C.9. Invite new members (only for companies)

Users that represent the company can invite new members to the GroupSpace. Invited users are older adults who can collaborate with a certain company.

D. Others Actions

With the term other actions, we assume those which are not mentioned above, like for example functionalities executed automatically by the system on the background context.

D.1. Update notifications through e-mail

Users are informed through e-mail about the progress of the job opportunity where he/she collaborates. This service will keep the users up to date.

D.2. Adapt the user interface manually

The user can change his/her interface appearance at any time. Those changes include the background colour and letter size.

D.3. Adapt user interface automatically

Through the data gathered from sensors and user interaction with the platform, this can decide an adaptation of the platform, this includes background colour, letter size and layout.

D.4. Match Skills automatically

Elders-Up! system matches automatically the needed skills from the job opportunity published in the platform with the ones that the older adult has in his/her profile. Job offers and skills provided by the user are saved in the Knowledge base from where Skills matching Service (SMS) module uses this information to find the most accurate match.

5.2 Functionality Priority

Table 4 lists all the functionalities specified, along with priorities for the implementation of these functions in the Elders-Up! System that has been taken into account in order to decide upon the final features.

Priority values: System Components:

- 1: Priority for the first prototype.
- 2: Priority for the final prototype.
- 3: Low priority –by the end of the project, but not essential.

Code	Functionality	Priority
A.1	User registration	1
A.1.1	Add personal information	1
A.1.2	Add Employment and Skills	1

A.1.3	Tell us about yourself	1
A.2	Update your profile	1
A.3	Sign In	1
A.4	Sign Out	1
B.1	Show the job opportunities	1
B.2	Create a GroupSpace	1
B.2.1	View Job opportunities	1
B.2.2	Edit Job opportunities	2
B.2.3	Delete Job opportunities	2
B.3	Enter to GroupSpace	1
B.4	Search for companies or experts	1
B.5	Tutor	2
B.6	Use Request module	1
B.6.1	Add task request	1
B.6.2	Respond to a task request	1
B.7	Use success case module	3
C.1	Manage GroupSpaces	1
C.2	Send messages	2
C.3	Send email	2
C.4	Start voice communication	2
C.5	Start video communication	3
C.6	Make appointments & track appointments	1
C.6.1	Create appointments	1
C.6.2	Accept invitations	1
C.7	Show a calendar common in the GroupSpace	1
C.8	Use file sharing module	2
C.8.1	Add and remove files	3
C.8.2	Add and remove folders	3

C.8.3	Edit Share files	2
C.9	Invite new members	1
D.1	Update notifications through email	1
D.2	Adapt the user interface manually	1
D.3	Adapt the user interface automatically	2
D.4	Skills matching automatically	1

Table 4. Functionality Priority

6 Use Cases

Based on the functionalities specified in the previous chapters, the most important use cases have been identified for the older adults and company users.

The following use cases are described using a common table schema. The main section is the Main Flow, where the use case is broken down into an ordered list of steps.

Some use cases are restricted to certain users. The following table describes each use case and its possible actors.

Use Case	Type of user
1. Create and configure an account	All users
2. Sign In	All users
3. Sign Out	All users
4. Update your profile	All users
5. Accept the job opportunity	Older adult user
6. Enter in a GroupSpace	All users
7. Search for a match	All users
8. Use Tutor	Older adult user
9. Create job opportunity	Company user
10. Response to a task request	All users
11. See successful collaborations	Company user
12. Create GroupSpace	Company user
13. Send messages or images	All users
14. Send emails	All users
15. Make appointments & track appointments	All users

16. Accept/Reject invitations	All users
17. Check shared calendar	All users
18. Add or remove files	All users
19. Edit shared files	All users
20. Invite new members	Company users
21. Adapt the user interface manually	All users
22. Adapt the user interface automatically	Older adult users (indirectly)

Table 5. Use cases and types of actors

6.1 Use Case: Create and configure an account

Use Case Number	1
Use Case Name	Create and configure account
Actors	End Users and Elders-Up! system.
Summary	Covers all the steps of creating and configuring an Elders-Up! account through the Elders-Up! app.
Trigger / intent	User enters the Elders-Up! system front-end.
Pre-conditions	<ul style="list-style-type: none"> The primary user is not yet registered in the system, but wishes to be a user of the Elder-Up! system. The primary user has an email address not registered in the system.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> The user enters a username and password. Elders-Up! system checks if there is no user id that uses the same username (or email). Enter <i>personal information</i>. Enter <i>Employment and Skills</i>. Enter <i>Tell us about yourself</i>. Elders-Up! system adds this new user to the Elders-Up! database. The Elders-Up! system shows to the user that user's account has been created successfully. The Elders-Up! System sends a confirmation e-mail to confirm the registration
Alternative flows	<ol style="list-style-type: none"> User is requested to choose another password and name. Password and name don't match the correct format User leaves the Elders-Up! system.
Exceptional flows	Operation fails: Account creation fails with error message.
Displayed information	Form to enter username, password, personal data, skill, CV file.
Post-conditions	The primary user has a configured account in the system, and may start using Elders-Up! collaboration platform.
Relation to other use cases	None of the rest of use cases can be performed unless this has been done successfully.

Table 6. Use Case of "Create and configuration account"

6.2 Use Case: Sign In

Use Case Number	2
-----------------	---

Use Case Name	Sign In
Actors	End Users and Elders-Up! system.
Summary	A registered user wants to use Elders-Up! system and the first step is the sign in.
Trigger / intent	When the user completes the sign in form and presses <i>Sign In</i> button.
Pre-conditions	<ul style="list-style-type: none"> User must be registered. User shouldn't be signed in.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> User completes the Sign In form. Presses the <i>Sign In</i> button. The Elders-Up! system checks introduced user and password. If successful, the user can see the main menu of Elders-Up!
Alternative flows	<ol style="list-style-type: none"> User completes the login form. Pushes the Login Button. The Elders-Up! system checks the user and password of the user. If not successful, the user can see over the login screen a notification with "The credentials are not valid"
Exceptional flows	
Displayed information	The user can watch main screen.
Post-conditions	The user can use the Elders-Up! system.
Relation to other use cases	This use case cannot be performed unless the use case " Create and configuration account " has been done successfully.

Table 7. Use Case of "User Login"

6.3 Use Case: Sign out

Use Case Number	3
Use Case Name	Sign out
Actors	End Users and Elders-Up! system.
Summary	A logged user wants to logout Elders-Up! system.
Trigger / intent	The user clicks or taps the <i>Sign Out</i> button
Pre-conditions	<ul style="list-style-type: none"> The user must be logged in
Flow of events: (Main Flow)	<ol style="list-style-type: none"> The user clicks or taps the <i>Sign out</i> button. Elders-Up! system removes the user session. The Elders-Up! system opens the Sign in screen and the user is signed out.
Alternative flows	
Exceptional flows	
Displayed information	User can see the Sign in screen
Post-conditions	User has to sign in if he wants to use the Elders-Up! system.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 8. Use Case of "Sign out"

6.4 Use Case: Update your profile

Use Case Number	4
Use Case Name	Update your profile
Actors	End Users and Elders-Up! system.
Summary	In his profile, the user can change information about him/her.
Trigger / intent	User activates <i>Profile</i> button.
Pre-conditions	<ul style="list-style-type: none"> The user must be signed in.

Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. The user clicks or taps <i>Profile</i> button. 2. A screen with his personal information is shown. 3. The user adds, updates, removes or changes outdated fields. 4. Clicks save button. 5. His profile is updated.
Alternative flows	<ol style="list-style-type: none"> 1. The user clicks or taps the <i>Profile</i> button. 2. A screen with his personal information is shown. 3. The user adds, removes or changes outdated fields. 4. Clicks cancel button or closes the window. 5. His profile is not updated.
Exceptional flows	
Displayed information	User can see his profile information.
Post-conditions	His updated profile can be seen by other users.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 9. Use Case of "Update your profile"

6.5 Use Case: Accept the job opportunity

Use Case Number	5
Use Case Name	Accept the job opportunity
Actors	End Users and Elders-Up! system.
Summary	User older adult accepts a new job opportunity in the search Company Screen.
Trigger / intent	User older adult pushes <i>Accept invitation</i> button.
Pre-conditions	<ul style="list-style-type: none"> • The user must be signed in.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User older adult clicks on search for a match. 2. User examine the matching companies and opportunities. 3. If the user agrees, he/she applies for an opportunity.
Alternative flows	<ol style="list-style-type: none"> 1. User older adult clicks or taps view opportunity from the Current Opportunities List. 2. He reads current opportunity information. 3. User older adult doesn't confirm interest in the job.
Exceptional flows	
Displayed information	A description of the current opportunity is shown for the senior expert, the company that offers it and information about the job.
Post-conditions	User older adult can see this new working agreement in his GroupSpace list once created by the company, and he/she is able to enter it.
Relation to other use cases	This case cannot be performed unless the user older adult is signed in. A company user has executed the use case <i>Create a new job opportunity</i> before.

Table 10. Use Case of "Accept the job opportunity"

6.6 Use Case: Enter in a GroupSpace

Use Case Number	6
Use Case Name	Enter in a workspace
Actors	End Users, Elders-Up! system.
Summary	Signed in user enters to a GroupSpace.
Trigger / intent	User is in the Dashboard. Starts this use case clicking link with other links to "your workspace"
Pre-conditions	<ul style="list-style-type: none"> • User must be signed in. • User is in the Elders-Up! <i>dashboard</i>

Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. Users sees the GroupSpace list the user is participating in. 2. User selects clicks or tabs on a GroupSpace. 3. The system shows the GroupSpace.
Alternative flows	
Exceptional flows	<ol style="list-style-type: none"> 1. The users does not click on a GroupSpace. 2. The elders-up! System does nothing
Displayed information	Shows a workspace with its visible content.
Post-conditions	User is in the selected workspace.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 11. Use Case of "Enter in a GroupSpace"

6.7 Use Case: Search for a match

Use Case Number	7
Use Case Name	Search
Actors	Senior Expert and Elders-Up! system
Summary	An older adult wants to find a company that needs help based on a matching percentage or a company user wants to find an older adult to collaborate.
Trigger / intent	User is in the Dashboard. Presses <i>Search for a match</i> button to find an older adult or a company.
Pre-conditions	<ul style="list-style-type: none"> • User must be signed in. • User is in the Elders-Up! <i>dashboard</i>
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User presses <i>Search for a match</i> button from the Dashboard. 2. The system shows all possible opportunities along with a matching percentage 3. User select the desired match 4. User selects a person or company clicking on <i>Show interest</i>.
Alternative flows	<ol style="list-style-type: none"> 1. User presses <i>Search</i> button from the Dashboard. 2. The system shows all possible opportunities along with a matching percentage 3. The system can't find a result. 4. User returns to <i>Main menu</i>.
Exceptional flows	<ol style="list-style-type: none"> 1. User presses <i>Search</i> button from the Dashboard. 2. User leaves search and returns to menu.
Displayed information	A list of users or companies.
Post-conditions	Older adult has found a company or a company has found an older adult.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 12. Use Case of "Search"

6.8 Use Case: Use Tutor

Use Case Number	8
Use Case Name	Use Tutor
Actors	End Users, Elders-Up! system.
Summary	A tutor menu located in the upper right corner provide assistance for the actions available in the platform
Trigger / intent	When the user begins use with the coaching module
Pre-conditions	<ul style="list-style-type: none"> • User must be signed in. • User is in the coaching module. • User is an older adult.
Flow of events:	<ol style="list-style-type: none"> 1. User starts Elders-Up! system.

(Main Flow)	<ol style="list-style-type: none"> Tutor is always active in the top of the dashboard. It shows relevant actions for the older adult and actions that should be completed. User interacts with tutor module completing some actions.
Alternative flows	<ol style="list-style-type: none"> User starts Elders-Up! system. Tutor module is always active in the top of the dashboard. It shows relevant actions for the older adult and actions that should be completed. User doesn't interact with tutor module completing some actions.
Exceptional flows	<ol style="list-style-type: none"> User starts Elders-Up! system. User disables the tutor, in this case it will stay disabled for future sessions.
Displayed information	Helping information and suggestions for the user. This is useful for a better user experience.
Post-conditions	If the user clicks on <i>the tutor icon</i> , new interfaces showing the tutor menu will appear.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 13. Use Case of "Use Tutor"

6.9 Use Case: Create job opportunity

Use Case Number	9
Use Case Name	Add request
Actors	End Users, Elders-Up! system.
Summary	In this use case new job opportunities are created by a company.
Trigger / intent	Company user pushes <i>add requests</i> button in the requests interface.
Pre-conditions	<ul style="list-style-type: none"> Sender and receiver of the request must be registered Both users are in the same GroupSpace.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> Company user click on <i>Search for an expert</i>. Company click on <i>Ass new search profile</i>. Company completes information regarding the new request. Company pushes <i>create opportunity</i> button. Elders-Up! system publish a new request that can be applied by senior experts.
Alternative flows	<ol style="list-style-type: none"> Company cancels the process.
Exceptional flows	
Displayed information	<p>An interface showing information for a new request:</p> <ul style="list-style-type: none"> -Title -Description -Skills sought -Start date -Competences -Languages
Post-conditions	The users can see the offered collaboration in the <i>Search for a match</i> module
Relation to other use cases	This case cannot be performed unless the user is signed in. User must have accepted a job opportunity.

Table 14. Use Case of "Add requests"

6.10 Use Case: Response to a task request

Use Case Number	10
Use Case Name	Response to a request

Actors	Senior expert, company, Elders-Up! system.
Summary	User older adult is assigned a task by another user or the company
Trigger / intent	User older adult pushes a request in the <i>Requests</i> menu and selects <i>Read request</i> .
Pre-conditions	<ul style="list-style-type: none"> Both users are registered. A user company has sent a request. Both users are in the same GroupSpace
Flow of events: (Main Flow)	<ol style="list-style-type: none"> Expert receives a task notification with a request. The expert clicks on the request inside his/her GroupSpace The expert accepts or rejects the request. The request is shown to be confirmed or rejected by the senior expert
Alternative flows	
Exceptional flows	
Displayed information	A notification of the task assigned along with a description of this task A pop-up menu when clicking on the task with possibility of accepting, rejecting or editing the task
Post-conditions	Both users have the tasks in the accepted requests menu.
Relation to other use cases	A company has sent the request in <i>Add request</i> use case.

Table 15. Use Case of "Response to a request"

6.11 Use Case: See success cases

Use Case Number	11
Use Case Name	See success cases
Actors	End Users, Elders-Up! system.
Summary	Some companies success cases of Elders-Up! system are described in this use case. Information about users or companies is not shown in this use case.
Trigger / intent	Company pushes the button <i>Success cases</i> in the dashboard.
Pre-conditions	<ul style="list-style-type: none"> User is signed in.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> Company clicks success cases button Elders-Up! system opens a new screen showing the anonymous success cases.
Alternative flows	
Exceptional flows	
Displayed information	A description of the most important success cases.
Post-conditions	User company has seen Elders-Up! system success cases.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 16. Use Case of "See success cases"

6.12 Use Case: Create GroupSpace

Use Case Number	12
Use Case Name	Create Job opportunities
Actors	Company, Elders-Up! system.
Summary	A user company wants to create a new GroupSpace with one or more seniors for a job opportunity previously created
Trigger / intent	Starts when the Company presses <i>Create a GroupSpace</i> in the dashboard.
Pre-conditions	<ul style="list-style-type: none"> User must be signed in. The Company requires particular skills.

Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. Company pushes <i>Create new GroupSpace</i> in the Main Menu. 2. The system opens a window with an application form to fill it with the name. 3. Company sees the recently created GroupSpace and can invite new members. 4. To invite new members, the company clicks on the "+" icon next to contacts. 5. The system sends an invitation to the invited members
Alternative flows	
Exceptional flows	
Displayed information	The system reports the GroupSpace creation to the user.
Post-conditions	GroupSpace is generated.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 17. Use Case of "Manage workspaces"

6.13 Use Case: Send messages or images

Use Case Number	13
Use Case Name	Send messages or images
Actors	End Users, Elders-Up! system.
Summary	All users can send messages or images to all other members of the workspace.
Trigger / intent	User clicks <i>Send</i> button in <i>Messages</i> interface in the group space.
Pre-conditions	<ul style="list-style-type: none"> • User has to be signed in. • Sender and receiver are in the same group space. • Sender is logged in.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User opens Messages screen. 2. Writes the message in <i>type message</i>. User can also add an image. 3. Clicks send. 4. Elders-Up! system adds the message to the messages in the group space.
Alternative flows	<ol style="list-style-type: none"> 1. User opens Messages screen. 2. Writes the message in <i>type message</i>. User can also add an image 3. The user doesn't click <i>Send</i> 4. The system doesn't send the message
Exceptional flows	
Displayed information	
Post-conditions	The message is added to the message list of the group space.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 18. Use Case of "Send messages or images"

6.14 Use Case: Send emails

Use Case Number	14
Use Case Name	Send emails
Actors	End Users, Elders-Up! system.
Summary	All users can send emails to other members of the group space.
Trigger / intent	User clicks opens the profile of a person in the group space.
Pre-conditions	<ul style="list-style-type: none"> • User has to be signed in.

	<ul style="list-style-type: none"> • Sender and receiver are in the same workspace.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. Selects a user to send him the mail. 2. Pushes the send email button. 3. An external mail client opens with a predefined recipient. This email client handles the rest of the interaction with the user.
Alternative flows	<ol style="list-style-type: none"> 1. Selects a user to send him the mail. 2. Pushes the send email button. 3. An external mail client opens with a predefined recipient. 4. The user closes the email client. No message has been send.
Exceptional flows	
Displayed information	
Post-conditions	Receiver has the new message in his email inbox.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 19. Use Case of "Send mails"

6.15 Use Case: Make appointments & track appointments

Use Case Number	15
Use Case Name	Make & track appointments
Actors	End Users, Elders-Up! system.
Summary	All users can create or follow appointments.
Trigger / intent	User pushes Appointments button in the workspace.
Pre-conditions	<ul style="list-style-type: none"> • User has to be signed in. • Both users have to be in the same workspace.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User clicks <i>Appointments</i> button. 2. User opens a form to generate an appointment. 3. User fills in the form. 4. User selects other users that can track the appointment. 5. Selects a date. 6. Saves the appointment. 7. The system registers the appointment and shows it in the Elders-Up! system calendar. 8. Other users can accept or refuse an appointment.
Alternative flows	
Exceptional flows	
Displayed information	Elders-Up! system shows in the calendar the registered appointment.
Post-conditions	The appointment was registered.
Relation to other use cases	This case cannot be performed unless the user is signed in.

Table 20. Use Case of "Make appointments & track appointments"

6.16 Use Case: Accept/Reject invitations

Use Case Number	16
Use Case Name	Accept/Reject invitations
Actors	End Users, Elders-Up! system.
Summary	User accepts or rejects appointment invitations
Trigger / intent	User can select a new appointment in <i>Agenda</i> ..
Pre-conditions	<ul style="list-style-type: none"> • User is signed in • Another user has created an appointment inviting him/her.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User selects an invitation. 2. A confirmation message is shown.

	<ol style="list-style-type: none"> 3. User accepts the appointment. 4. The updated appointment is stored in agenda.
Alternative flows	<ol style="list-style-type: none"> 1. User selects an invitation. 2. A confirmation message is shown. 3. User rejects the appointment.
Exceptional flows	
Displayed information	Information about the invitation and two buttons; OK and cancel.
Post-conditions	The system registers the accepted or refused invitation.
Relation to other use cases	An appointment must have been created in the "Make & track appointments" use case.

Table 21. Use Case of "Accept/Reject invitations"

6.17 Use Case: Check shared calendar

Use Case Number	17
Use Case Name	Check shared calendar
Actors	End Users, Elders-Up! system.
Summary	User sees his appointments in the calendar.
Trigger / intent	User pushes Appointments button.
Pre-conditions	<ul style="list-style-type: none"> • User has to be logged in.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User pushes <i>Appointments</i>. 2. Elders-Up! system opens Agenda. 3. The system shows the calendar shared in the workspace.
Alternative flows	
Exceptional flows	
Displayed information	A calendar with accepted and invitations of new appointments is displayed.
Post-conditions	User is in a window that allows him to see the shared calendar.
Relation to other use cases	

Table 22. Use Case of "See the common calendar"

6.18 Use Case: Add or remove files

Use Case Number	18
Use Case Name	Add or remove files
Actors	End Users, Elders-Up! system.
Summary	User adds or removes files in <i>File Sharing</i> screen. In the file sharing system a folder for each project has been created. Every type of user can add or remove a file.
Trigger / intent	User pushes add file button to create a new file in a shared project. User selects a file and pushes <i>remove</i> button to delete a file from a shared project.
Pre-conditions	<ul style="list-style-type: none"> • User has to be signed in • To add or open files of a project he has to be part of it.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User pushes <i>file sharing</i>. 2. Elders-Up! system opens <i>File Sharing</i> interface. 3. User pushes <i>add new file</i>. 4. Elders-Up! system opens a window where the user can search and select the file he wants to upload. 5. User selects the file and clicks upload. 6. Elders-Up! system creates the file in the folder.
Alternative flows	<ol style="list-style-type: none"> 1. User selects a file. 2. Pushes delete button.

	<ol style="list-style-type: none"> 3. Elders-Up! system opens a confirmation message to delete the file. 4. User accepts to delete the file. 5. Elders-Up! system removes the file from the folder.
Exceptional flows	
Displayed information	A folders and files structure.
Post-conditions	Files were added/deleted from the shared folder.
Relation to other use cases	

Table 23. Use Case of "Add or remove files"

6.19 Use Case: Edit shared files

Use Case Number	19
Use Case Name	Edit shared files.
Actors	End Users, Elders-Up! system.
Summary	User edits a file from the <i>file sharing</i> interface. Every user can edit a file of a shared workspace.
Trigger / intent	User double-clicks a file (or press the button "Open file" when the file is selected)
Pre-conditions	<ul style="list-style-type: none"> • User has to be signed in • To edit a file of a project he has to be part of it.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User double-clicks a file. 2. Elders-Up! system opens the file. 3. User edits the file. 4. User saves the file. 5. Elders-Up! system opens a confirmation message to save the new file. 6. User accepts to save the changes. 7. Elders-Up! system saves the changes in the file.
Alternative flows	<ol style="list-style-type: none"> 1. User double-clicks a file. 2. Elders-Up! system opens the file. 3. User edits the file. 4. User saves the file. 5. Elders-Up! system opens a confirmation message to save the new file. 6. User refuses to save the changes.
Exceptional flows	-
Displayed information	"Last modified by:" and "Modified on:" are updated with new values.
Post-conditions	The file is updated in the system.
Relation to other use cases	

Table 24. "Use Case of "Edit shared files"

6.20 Use Case: Invite new members

Use Case Number	20
Use Case Name	Invite new members
Actors	Company, Elders-Up! system.
Summary	Company can invite new members to his group. Those invited users are older adults who can collaborate with the company.
Trigger / intent	User pushes invite new members' button.
Pre-conditions	<ul style="list-style-type: none"> • User has to be signed in as a company.
Flow of events:	<ol style="list-style-type: none"> 1. User pushes <i>invite new members</i>.

(Main Flow)	<ol style="list-style-type: none"> 2. A list with users is shown. 3. Selects a user. 4. Pushes <i>add new member</i>. 5. Elders-Up! system sends a confirmation message to the user.
Alternative flows	
Exceptional flows	
Displayed information	Elders-Up! system shows the message: "A new request was sent".
Post-conditions	A new member's request is sent to the selected member.
Relation to other use cases	

Table 25. Use Case of "Invite new members"

6.21 Use Case: Adapt the user interface manually

Use Case Number	21
Use Case Name	Adapt the user interface manually
Actors	End Users, Elders-Up! system.
Summary	User opens settings and changes the graphical user interface options.
Trigger / intent	User clicks on <i>Settings</i> .
Pre-conditions	<ul style="list-style-type: none"> • User has signed in. • User must be an older adult user.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. User clicks <i>Settings</i>. 2. User modifies view settings 3. Elders-Up! system shows the graphical user interface with the new settings. 4. User confirms changes.
Alternative flows	
Exceptional flows	
Displayed information	A Settings menu interface is displayed.
Post-conditions	The system changes the appearance of the graphical user interfaces.
Relation to other use case	

Table 26. Use Case of "Adapt the user interface manually"

6.22 Use Case: Adapt the user interface automatically

Use Case Number	22
Use Case Name	Adapt the user interface automatically
Actors	Elders-Up! system, older adult
Summary	In this use case, Elders-Up! system collects data from the user's ambient and situation adapting user's UI to the detected conditions.
Trigger / intent	
Pre-conditions	<ul style="list-style-type: none"> • User has installed Elders-Up! SDC. in his device. • SDC has registered user's valid ID. • User is logged in Elders-Up! application.
Flow of events: (Main Flow)	<ol style="list-style-type: none"> 1. Elders-Up! system collects user data 2. Elders-Up! system checks collected data to detect potential problems 3. Elders-Up! system Detects a problem and propose a change of the graphical user interface 4. The older adult accepts the change 5. The system automatically updates interfaces.
Alternative flows	<ol style="list-style-type: none"> 1. Elders-Up! system collects user data

	2. User interface is not updated.
Exceptional flows	
Displayed information	
Post-conditions	User interface has been automatically adapted for the user.
Relation to other use case	

Table 27. Use Case of "Adapt the user interface automatically"

7 API Specification

The complete API specification is included in D4.3 Final Elders-Up! Integrated prototype. In this document will find the APIs developed for the communication between the different modules and servers. Firstly, the API created for the communications between EUP_SERVER_1 and the different modules is shown. After this, the APIs developed for the communication of EUP_SERVER_2 with the different components is shown. Finally we will find the APIs developed for the following modules: SMS, SRC, Tutor and SDC.

Commented [IMA1]: Maybe by making a brief summary of what is included in D4.3 would offer a better view of the API. I know that it has been moved there, but API appears in the title of D2.7.

8 List of Figures

Figure 1. The six stages towards final functional requirements in the Elders-Up user-centric system design methodology	7
Figure 2: Elders-Up! application structure	10
Figure 3. Personal data	11
Figure 4. Senior user skills	12
Figure 5. Motivatoion and availability	12
Figure 6. Dashboard senior expert	14
Figure 7. SMS GUI	15
Figure 8. ISEU GUI with simplified obligatory fields	17
Figure 9. Search GUI with available users	17
Figure 10. Opportunity creation GUI	18
Figure 11. Group Messages	19
Figure 12. Files sharing system	19
Figure 13: Self-Reporting Collection structure	20
Figure 14. SRC GUI	22
Figure 15: Elders-Up! DM System	23
Figure 16. Adaptation Decision-Maker design	24
Figure 17. Example of adaptation solution and associated ICAW UI	24
Figure 18. Final module logical architecture and relation to other Elders-Up! Components	25
Figure 19. SME GUI for entering skills requested for a job opportunity	26
Figure 20. Knowledge base logical connection to other components with in/out flow of the information direction between modules	27
Figure 21. Email notification sent by the platform	28
Figure 22. Email notification sent by the platform (2)	28
Figure 23. Tutor help button	28
Figure 24. General architecture of the tutor module	29
Figure 25. Tutor main menu	30
Figure 26. MediForm main page	30

Figure 27. MediForm platform schema 31
Figure 28. Final Elders-Up! global system architecture 32

9 List of tables

Table 1. Deliverables and milestones from P1.0 to fina prototype	7
Table 2. End user Requirements	13
Table 3. End User Requirements (2)	16
Table 4. Functionality Priority	43
Table 5. Use cases and types of actors	45
Table 6. Use Case of "Create and configuration account"	45
Table 7. Use Case of "User Login"	46
Table 8. Use Case of "Sign out"	46
Table 9. Use Case of "Update your profile"	47
Table 10. Use Case of "Accept the job opportunity"	47
Table 11. Use Case of "Enter in a GroupSpace"	48
Table 12. Use Case of "Search"	48
Table 13. Use Case of "Use Tutor"	49
Table 14. Use Case of "Add requests"	49
Table 15. Use Case of "Response to a request"	50
Table 16. Use Case of "See success cases"	50
Table 17. Use Case of "Manage workspaces"	51
Table 18. Use Case of "Send messages or images"	51
Table 19. Use Case of "Send mails"	52
Table 20. Use Case of "Make appointments & track appointments"	52
Table 21. Use Case of "Accept/Reject invitations"	53
Table 22. Use Case of "See the common calendar"	53
Table 23. Use Case of "Add or remove files"	54
Table 24. "Use Case of "Edit shared files"	54
Table 25. Use Case of "Invite new members"	55
Table 26. Use Case of "Adapt the user interface manually"	55
Table 27. Use Case of "Adapt the user interface automatically"	56