

**REBUILD**  
ICT-enabled integration facilitator and life rebuilding guidance  
*Project start date: 01/01/2019 | Duration: 36 months*

# Deliverable: D5.2 REBUILD Pilot Platform first prototype

DUE DATE OF THE DELIVERABLE: 31-03-2020

ACTUAL SUBMISSION DATE: 29-05-2020

<b>Project</b>	REBUILD – ICT-enabled integration facilitator and life rebuilding guidance
<b>Call ID</b>	H2020-SC6-MIGRATION-2018-2019-2020 – DT-MIGRATION-06-2018
<b>Work Package</b>	<i>WP5 – REBUILD Reference Architecture</i>
<b>Work Package Leader</b>	<i>Engineering Ingegneria Informatica S.p.A.</i>
<b>Deliverable Leader</b>	<i>Engineering Ingegneria Informatica S.p.A.</i>
<b>Deliverable coordinator</b>	Antonio Filograna (ENG) – <a href="mailto:antonio.filograna@eng.it">antonio.filograna@eng.it</a>
<b>Deliverable Nature</b>	Demonstrator
<b>Dissemination level</b>	Public (PU)
<b>Version</b>	1.6
<b>Revision</b>	Final

# 1. DOCUMENT INFO

## 1. AUTHORS

Author name	Organization	E-Mail
Antonio Filograna	ENG	<a href="mailto:antonio.filograna@eng.it">antonio.filograna@eng.it</a>
Thodoris Semertzidis	CERTH	<a href="mailto:theosem@iti.gr">theosem@iti.gr</a>
Gustavo Hernández	UPM	<a href="mailto:ghp@gatv.ssr.upm.es">ghp@gatv.ssr.upm.es</a>

## 2. DOCUMENT HISTORY

Version #	Author name	Date	Changes
0.1	Antonio Filograna (ENG)	15-01-2020	Table of Content definition
0.2	Antonio Filograna (ENG) Gustavo Hernández (UPM) Thodoris Semertzidis (CERTH)	29-01-2020	Use case Diagram definition
0.3	Antonio Filograna (ENG) Gustavo Hernández (UPM) Thodoris Semertzidis (CERTH)	05-02-2020	Sequence Diagram definition
0.4	Antonio Filograna (ENG)	12-02-2020	REBUILD Architecture recap
0.5	Antonio Filograna (ENG)	26-03-2020	Integration Use Case first draft
0.6	Antonio Filograna (ENG) Gustavo Hernández (UPM) Thodoris Semertzidis (CERTH)	10-04-2020	Infrastructure requirements
0.7	Antonio Filograna (ENG)	15-04-2020	Data Model first draft
0.8	Antonio Filograna (ENG) Gustavo Hernández (UPM) Thodoris Semertzidis (CERTH)	27-04-2020	Integration Use Case finalisation
0.9	Gustavo Hernández (UPM)	30-04-2020	Interaction architecture and APIs first draft
1.0	Antonio Filograna (ENG) Alessandro Caforio (UNINET)	04-05-2020	REBUILD APP UI
1.1	Antonio Filograna (ENG) Gustavo Hernández (UPM) Thodoris Semertzidis (CERTH)	07-05-2020	Data Model final version
1.2	Antonio Filograna (ENG) Gustavo Hernández (UPM) Thodoris Semertzidis (CERTH)	11-05-2020	Interaction architecture and APIs final version
1.3	Antonio Filograna (ENG)	13-05-2020	Executive Summary Conclusion
1.4	Demertzis Dimitris, OMNES Christos Gkelinos, OMNES Luigi Laura, UNINETTUNO	20-05-2020	Internal Review
1.5	Antonio Filograna (ENG)	26-05-2020	Final version to be submitted
1.6	Maria Amata Garito (UNINETTUNO)	29-05-2020	Submission to EC



Re\_Build

ICT-enabled  
integration facilitator  
and life rebuilding guidance

This project has received funding from the European  
Union's Horizon 2020 research and innovation  
programme under grant agreement No 822215



### 3. DOCUMENT DATA

<b>Keywords</b>	<i>Demonstrator, Data Model, REBUILD Platform</i>
<b>Editor Address data</b>	Name: Antonio Filograna Partner: ENG Address: via per Monteroni, 73100 Lecce, Italy Phone: +393331229818 Email: <a href="mailto:antonio.filograna@eng.it">antonio.filograna@eng.it</a>
<b>Delivery Date</b>	29-05-2020
<b>Peer Review</b>	<i>Demertzis Dimitris, OMNES Christos Gkelinos, OMNES Luigi Laura, UNINETTUNO</i>

## 2. EXECUTIVE SUMMARY

---

This report includes the results of project tasks T5.2 "**Platform Architecture**" and T5.3 "**User Interface**". The aim of this document is to describe the activities performed to develop and set up the REBUILD Platform. Starting from the REBUILD high-level Architecture, the REBUILD Platform is its concrete instantiation. In doing so, the **Use Case and Sequence Diagrams** for each component were defined to understand the behaviour and the functionalities of all components.

The functionalities provided by each component led to the **Integration Use Cases** that aim to evaluate the integration degree of the whole platform. The Interaction Use Cases (IUC) describe the foreseen behaviour of the system during the interaction among components. As final test of the Platform, each IUC will be checked if it works properly, and in case correct the errors.

The REBUILD Platform will have a single **deployment** with three different instantiations, one per each pilot. If some pilot needs a customized deploy of the platform, this will be provided taking into account the new requirements. To guarantee the right computational resources the **Infrastructure Requirements** were defined and an installation guide for each component was edited.

To facilitate the installation and deployment of each component and the overall platform, the **dockerization** of them was performed. The description of instruction for each component has been published on the project Github account.

One of the most important pieces of the REBUILD Platform is the definition of its **Data Model**. Starting from the Use Case Scenarios defined in D5.1 [1] and D2.5 [2], all the information needed to the operating of the platform was collected and the Data Model was defined. The REBUILD Data Base was built following the Data Model.

In the end, the mock-ups of the **REBUILD mobile Application** were depicted and the development of the whole App, integrating each component and its related functionalities, was started.

## 3. TABLE OF CONTENTS

<b>1. DOCUMENT INFO</b>	<b>2</b>
1. AUTHORS	2
2. DOCUMENT HISTORY	2
3. DOCUMENT DATA	3
<b>2. EXECUTIVE SUMMARY</b>	<b>4</b>
<b>3. TABLE OF CONTENTS</b>	<b>5</b>
1. INDEX OF TABLES	6
2. INDEX OF FIGURES	6
<b>1 INTRODUCTION</b>	<b>7</b>
<b>2 REBUILD PLATFORM</b>	<b>8</b>
2.1 REBUILD ARCHITECTURE RECAP	8
2.2 INTERACTION ARCHITECTURE AND APIS	9
2.3 USE CASE AND SEQUENCE DIAGRAMS	11
2.3.1 <i>Request info from ChatBot using text</i>	12
2.3.2 <i>Request info from ChatBot using video</i>	13
2.3.3 <i>Communicate through Pictograms</i>	14
2.3.4 <i>Build User Profile</i>	15
2.3.5 <i>Receive suggestion based on Matchmaking</i>	16
2.3.6 <i>Authenticate</i>	17
<b>3 INTEGRATION PLAN</b>	<b>19</b>
3.1 INTEGRATION USE CASES	21
3.2 INFRASTRUCTURE REQUIREMENTS AND SETUP	24
3.2.1 <i>Data Base installation and setup</i>	24
3.2.2 <i>ChatBot installation and setup</i>	24
3.2.3 <i>Task Solver installation and setup</i>	25
3.2.4 <i>Task Solver Mobile App installation and setup</i>	25
3.2.5 <i>User Profiling installation and setup</i>	25
3.2.6 <i>Matchmaking Engine installation and setup</i>	25
3.2.7 <i>Recommendation System installation and setup</i>	25
<b>4 REBUILD DATA MODEL</b>	<b>26</b>
4.1 ISA <sup>2</sup> VOCABULARIES	28
4.2 REBUILD SPECIFIC DATA MODEL	28
<b>5 REBUILD USER INTERFACES</b>	<b>32</b>
5.1 USER INTERFACE MODEL	32
5.2 USER INTERFACE LAYOUT	34
<b>6 CONCLUSION</b>	<b>37</b>
<b>7. REFERENCE</b>	<b>38</b>

## 1. INDEX OF TABLES

Table 1: API1 end-points .....	9
Table 2: Integration Use Cases .....	21
Table 3: Infrastructure Requirements .....	24
Table 4: ISA2 Vocabularies .....	28
Table 5: REBUILD specific Data Model .....	28

## 2. INDEX OF FIGURES

Figure 1: REBUILD Interaction Architecture .....	9
Figure 2: Use Case Diagram.....	12
Figure 3: REBUILD Platform Deployment Diagram.....	19
Figure 4: The REBUILD Data Model .....	27
Figure 5: Private vs Wide approach .....	32
Figure 6: Dot grid model.....	33
Figure 7: REBUILD mobile APP UI approach.....	34
Figure 8: REBUILD mobile APP - main page .....	35
Figure 9: Ask a question through text or pictograms .....	36
Figure 10: Make a question recording a video .....	36

# 1 INTRODUCTION

---

The project REBUILD aims at **improving migrants and refugees' inclusion** through the provision of a toolbox of ICT-based solutions aimed to enhance both the effectiveness of the services provided by local public administration and organizations, and the life quality of the migrants.

This project follows a **user-centred and participated design approach**, aiming at addressing properly real target users' needs, ethical and cross-cultural dimensions, and at monitoring and validating the socio-economic impact of the proposed solution. Both target groups (immigrants/refugees and local public services providers) will be part of a continuous design process; users and stakeholders' engagement is a key success factor addressed both in the Consortium composition and in its capacity to engage relevant stakeholders external to the project. Users will be engaged since the beginning of the project through interviews and focus groups; then will be part of the application design, participating in three **Co-Creation Workshops** organized in the three main piloting countries: Italy, Spain and Greece, chosen for their being the "access gates" to Europe for main immigration routes. Then again, in the 2<sup>nd</sup> and 3<sup>rd</sup> years of the project, users' engagement in Test and Piloting events in the three target countries, will help the Consortium fine-tuning the REBUILD ICT toolbox before the end of the project.

The key technology solutions proposed are:

- **GDPR-compliant** migrants' integration related background information gathering with user consent and anonymization of personal information;
- **AI-based profile analysis** to enable both personalized support and policy making on migration-related issues;
- **AI-based needs matching tool**, to match migrant needs and skills with services provided by local authorities in EU countries and labour market needs at local and regional level;
- a **Digital Companion** for migrants enabling personalized two-way communication using **chatbots** to provide them smart support for easy access to local services (training, health, employment, welfare, etc.) and assessment of the level of integration and understanding of the new society, while providing to local authorities data-driven, easy to use decision supporting tools for enhancing capacities and effectiveness in service provision.

## 2 REBUILD PLATFORM

The first version of the **REBUILD Architecture** was released at M12 and will be updated after the first phase of experimentation. This Chapter describes the REBUILD Platform, the real implementation of the REBUILD Architecture, that is the set of back-end and front-end elements.

The following section was taken entirely from D5.1 [1] where the REBUILD Architecture was defined, just to give the reader the whole overview of the components and their relation.

### 2.1 REBUILD ARCHITECTURE RECAP

One of the outcomes of the REBUILD project is a smartphone application (RB App). The RB App offers two types of content: free content and content displayable after registration and profiling of user. Migrants can use the free content whenever they want, but if they need personalised services they have to register. The REBUILD platform profiles the users compliant with the GDPR, storing each kind of the user data (personal and not-personal data) in the **RB Data Base** by using cryptographic techniques to assure the privacy and protection of data. The **User Profiling** is in charge of building a multi-spatial vector to represent the profile of the user.

#### High-level Logic view

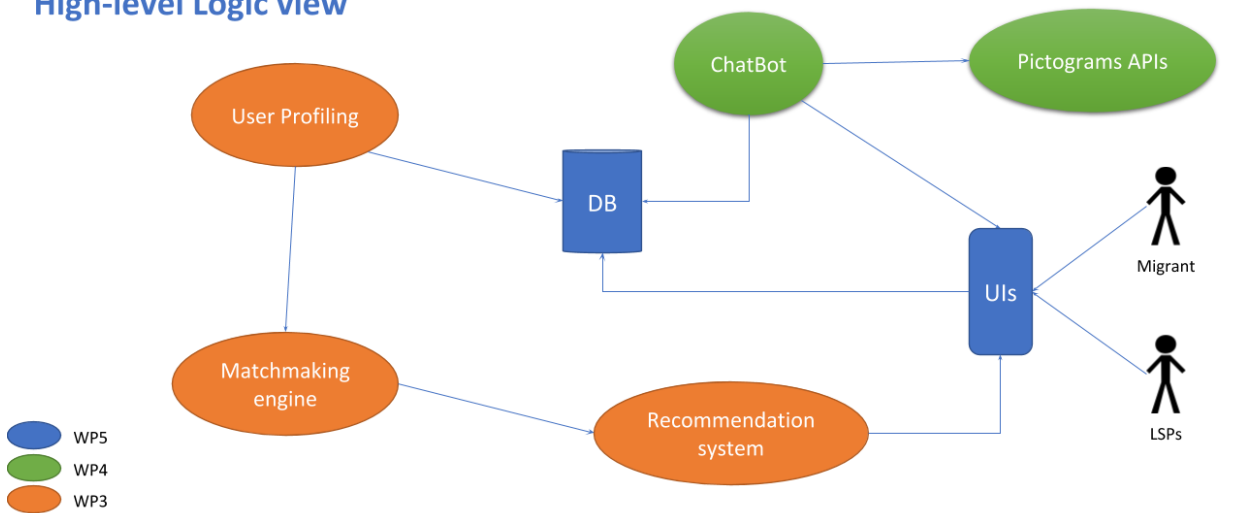


Figure 1 - REBUILD High-level Logic Architecture

REBUILD App is able to provide end-users with personalised suggestion for a specific service. Users can talk with the **ChatBot** in order to receive help and suggestion on specific needs. Migrants can interact with the ChatBot in three different modalities: writing, speaking and use the pictograms provided by **Pictograms APIs**. The ChatBot uses the **Recommendation System**, that leverages the user profile vector and the Matchmaking System to analyse the data coming from different kind of data (e.g., personal data, educational data, skills, user desiderata, etc.), to provide suggestion to the end-user. The **Matchmaking System** works on the basis of the data provided by migrants and LSPs, for example in the scenario of Job seeking this component is able to match the skill of the migrants, their past employments with the more requested job profiles provided by LSPs.

REBUILD Architecture was designed to meet all the needs coming from the Use Case Scenarios without having a specific purpose. The idea is to have a **modular platform** in order to integrate and use specific technical components according to the specific requested service. This way to proceed guarantees the extensibility of the platform by adding new technical components with a limited effort, fostering the **interoperability** and **scalability**.



## 2.2 INTERACTION ARCHITECTURE AND APIS

The following **Interaction Architecture** (Figure 1) describes how the front-end components interact with the back-end ones. The logical procedure is the following: user interfaces interact with the system via an API (API1) that will broker the events. The API1 will process the login services and forward the information through the rest of modules, acting as a sort of reverse proxy.

This API will permit the integration of all REBUILD toolbox. There are some APIs on top some modules, just to guarantee internal interconnectivity (i.e. Skill matching and Recommendation interact often). Also, User profiling might not have any type of interaction except for the skills matching.

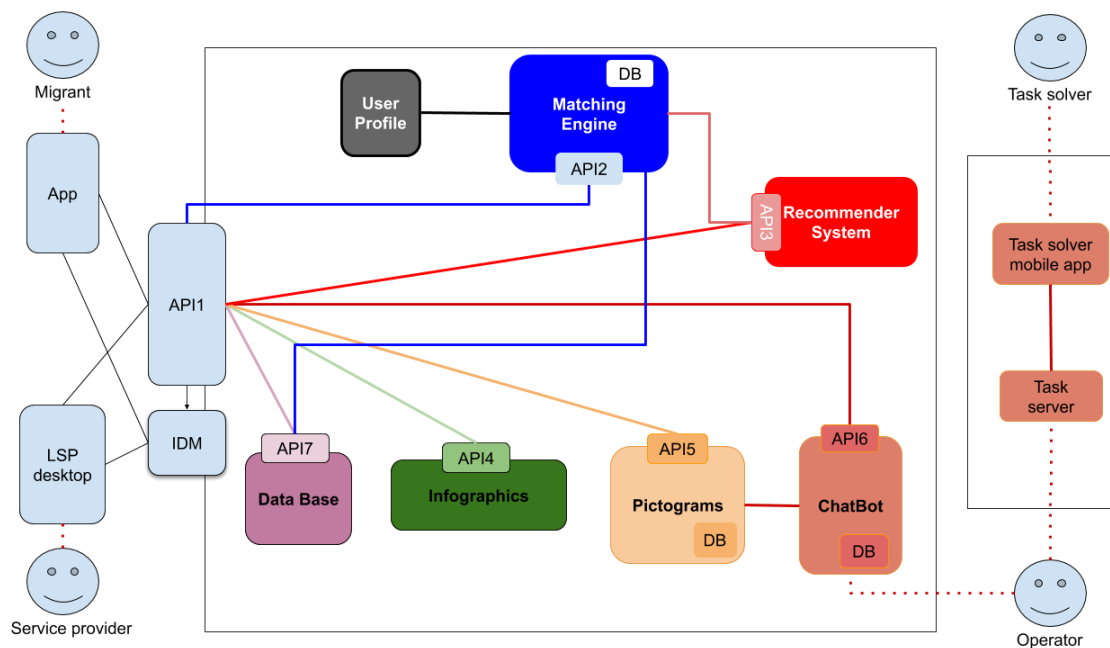


Figure 1: REBUILD Interaction Architecture

Therefore, the API1 will contain the following endpoints.

Table 1: API1 end-points

Type	API 1 endpoint	Description	Target
post	/oauth2/token	Retrieve the token	User
post	/oauth2/token	Refresh token	User
get	/api/client/lsp	Consult LSPs	(Any)
get	/api/client/lsp/{id}	Read detail about an LSP by identifier	(Any)
get	/api/client/lsp/{id}/courses	Read courses of an LSP by identifier	(Any)
get	/api/client/lsp/{id}/courses/{courseID}	Read single course of an LSP by identifiers	(Any)
post	/api/admin/lsp	Insert new LSP	Admin
put	/api/admin/lsp/{id}	Update an LSP by identifier	Admin
delete	/api/admin/lsp/{id}	delete an LSP by identifier	Admin
put	/api/admin/lsp/{id}/courses	Add new Course to LSP by identifier	Admin
delete	/api/admin/lsp/{id}/courses/{courseID}	Delete course from LSP by identifiers	Admin
get	/api/admin/migrants	Read all migrants' PersonalData	Admin
post	/api/admin/migrants	Create new migrant PersonalData	Admin
get	/api/admin/migrants/{id}	Read migrant's PersonalData by identifier	Admin
put	/api/admin/migrants/{id}	Update migrant's PersonalData by identifier	Admin

Re Build

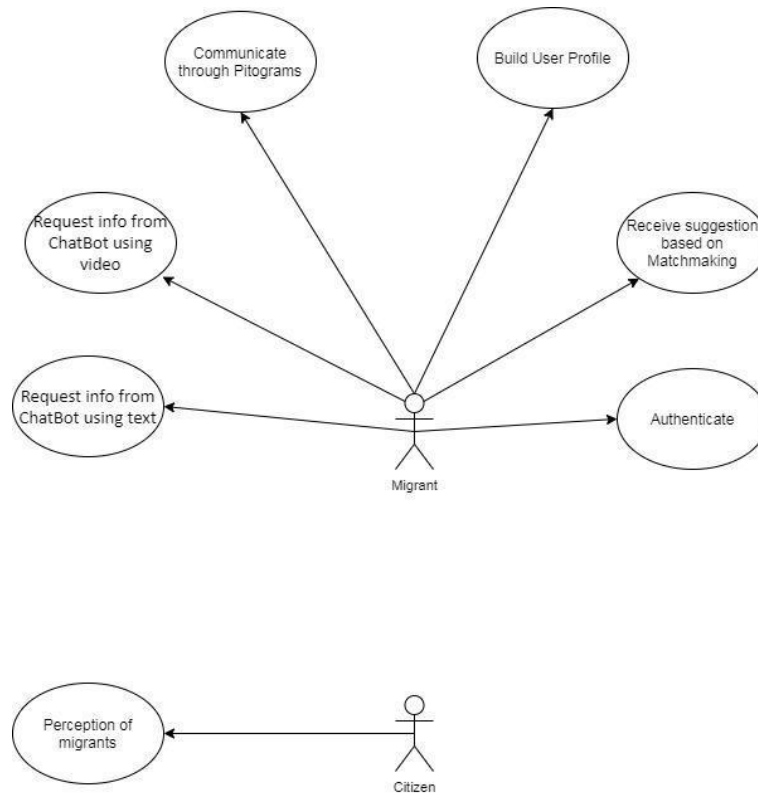
delete	/api/admin/migrants/{id}	Delete migrant's PersonalData by identifier	Admin
get	/api/admin/migrants/{id}/certifications	Read migrant's certifications by identifier	Admin
put	/api/admin/migrants/{id}/certifications	Add new certification to migrant by identifier	Admin
get	/api/admin/migrants/{id}/certifications/{id}	Read migrant's certification by identifiers	Admin
delete	/api/admin/migrants/{id}/certifications/{id}	Delete migrant's certification by identifiers	Admin
get	/api/admin/migrants/{id}/educationexperiences	Read migrant's education experiences by identifier	Admin
put	/api/admin/migrants/{id}/educationexperiences	Add new education experience to migrant by identifier	Admin
get	/api/admin/migrants/{id}/educationexperiences/{id}	Read migrant's education experience by identifiers	Admin
delete	/api/admin/migrants/{id}/educationexperiences/{id}	Delete migrant's education experience by identifiers	Admin
get	/api/admin/migrants/{id}/workexperiences	Read migrant's work experiences by identifier	Admin
put	/api/admin/migrants/{id}/workexperiences	Add new work experience to migrant by identifier	Admin
get	/api/admin/migrants/{id}/workexperiences/{id}	Read migrant's work experience by identifiers	Admin
delete	/api/admin/migrants/{id}/workexperiences/{id}	Delete migrant's education experience by identifiers	Admin
get	/api/admin/migrants/{id}/otherlanguages	Read migrant's languages by identifier	Admin
put	/api/admin/migrants/{id}/otherlanguages	Add new language to migrant by identifier	Admin
get	/api/admin/migrants/{id}/otherlanguages/{id}	Read migrant's language by identifiers	Admin
delete	/api/admin/migrants/{id}/otherlanguages/{id}	Delete migrant's language by identifiers	Admin
get	/api/admin/migrants/{id}/personaldocuments	Read migrant's personal documents by identifier	Admin
put	/api/admin/migrants/{id}/personaldocuments	Add new personal document to migrant by identifier	Admin
get	/api/admin/migrants/{id}/personaldocuments/{id}	Read migrant's personal document by identifiers	Admin
delete	/api/admin/migrants/{id}/personaldocuments/{id}	Delete migrant's personal document by identifiers	Admin
get	/api/migrants	Read migrant's PersonalData	Migrant
post	/api/migrants	Create migrant's PersonalData	Migrant
put	/api/migrants	Update migrant's PersonalData	Migrant
delete	/api/migrants	Delete migrant's PersonalData	Migrant
get	/api/migrants/certifications	Read migrant's certifications	Migrant
put	/api/migrants/certifications	Add new certification to migrant	Migrant
get	/api/migrants/certifications/{id}	Read migrant's certification by identifier	Migrant
delete	/api/migrants/certifications/{id}	Delete migrant's certification by identifiers	Migrant
get	/api/migrants/educationexperiences	Read migrant's education experiences	Migrant
put	/api/migrants/educationexperiences	Add new education experience to migrant	Migrant
get	/api/migrants/educationexperiences/{id}	Read migrant's education experience by identifiers	Migrant
delete	/api/migrants/educationexperiences/{id}	Delete migrant's education experience by identifiers	Migrant
get	/api/migrants/workexperiences	Read migrant's work experiences	Migrant
put	/api/migrants/workexperiences	Add new work experience to migrant	Migrant
get	/api/migrants/workexperiences/{id}	Read migrant's work experience by	Migrant

		identifiers	
delete	/api/migrants/workexperiences/{id}	Delete migrant's education experience by identifiers	Migrant
get	/api/migrants/otherlanguages	Read migrant's languages	Migrant
put	/api/migrants/otherlanguages	Add new language to migrant	Migrant
get	/api/migrants/otherlanguages/{id}	Read migrant's language by identifiers	Migrant
delete	/api/migrants/otherlanguages/{id}	Delete migrant's language by identifiers	Migrant
get	/api/migrants/personaldocuments	Read migrant's personal documents	Migrant
put	/api/migrants/personaldocuments	Add new personal document to migrant	Migrant
get	/api/migrants/personaldocuments/{id}	Read migrant's personal document by identifiers	Migrant
get	/api/lsp	Read LSP details	LSP
post	/api/lsp	Insert new LSP details	LSP
put	/api/lsp	Update the LSP	LSP
delete	/api/lsp	delete the LSP	LSP
get	/api/lsp/courses	Read LSP's courses	LSP
put	/api/lsp/courses	Add new Course to LSP	LSP
get	/api/lsp/courses/{courseID}	Read single course of an LSP by identifier	LSP
delete	/api/lsp/courses/{courseID}	Delete course from LSP	LSP
post	/service/job	Post a job offer	Service skills based matching
put	/service/job/{id}	Validate a Job offer	Service skills based matching
get	/service/job	Get all jobs available	Service skills based matching
post	/service/recommendation	Launch the recomender engine	Recommendation system
get	/service/recommendation/{tag}/{id}	Retrieve a recommendation by tag and user identifier	Recommendation system
post	/service/infographics	Retrieve the information from the infographics on a particular topic	Infographics
post	/service/pictograms	Create a pictogram	Pictograms
put	/service/updatePic	Update a pictogram	Pictograms
get	/service/pictogram/{id}	retrieve	Pictograms
post	service/keys	Create a new Key	Db operations
get	service/key/{id}	Key consult	Db Operations
delete	service/key/{id}	Delete a Key	Db Operations
post	/service/updateProfile	A service wants to update information on a particular profile	Db operations
post	/service/chat/message	Send a new message	Chat operations
get	/service/chat/message/{id}	Read a message	Chat operations
socket.io	/service/chat/notifications/{user}	Get notification that a message for the user exists in the database	Chat operations
get	/service/chat/history/{user}	Get the chat history of a user	Chat operations
post	/service/chat/upload	Upload a media file (image/video)	Chat operations

## 2.3 USE CASE AND SEQUENCE DIAGRAMS

In this section an overview is given of the main use cases that will be addressed in the course of the project. For each use case, it will be clearly identified the *participating actor*, i.e., the role responsible to trigger the case and, at the same time, the beneficiary of the actions taken within the case. Use cases are labelled with names which in some cases recall the action taken by the actor to trigger the use case, in some others they identify the

primary action taken by the REBUILD framework. In Figure 2 a **UML use case diagram** of the main cases addressed in the project is depicted:

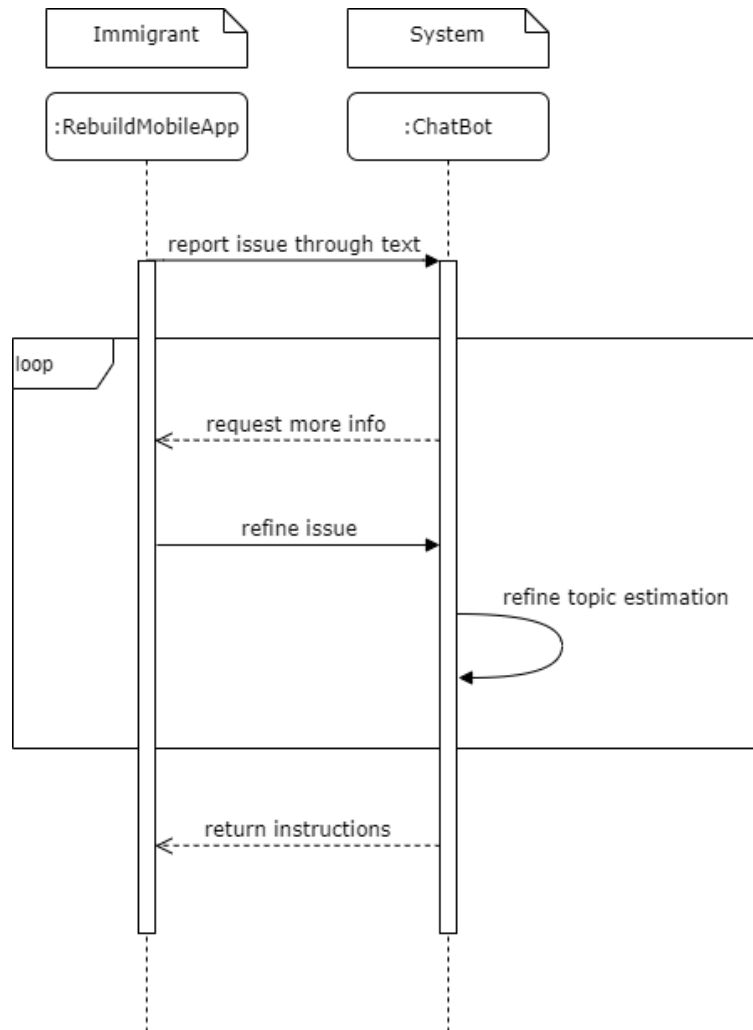


*Figure 2: Use Case Diagram*

In the following, we report a list of **UML sequence diagrams** corresponding to each of the use cases depicted in Figure 2. Sequence diagrams highlight how the REBUILD framework’s components interact to each other’s in order to carry out a specific action requested by the participating actor.

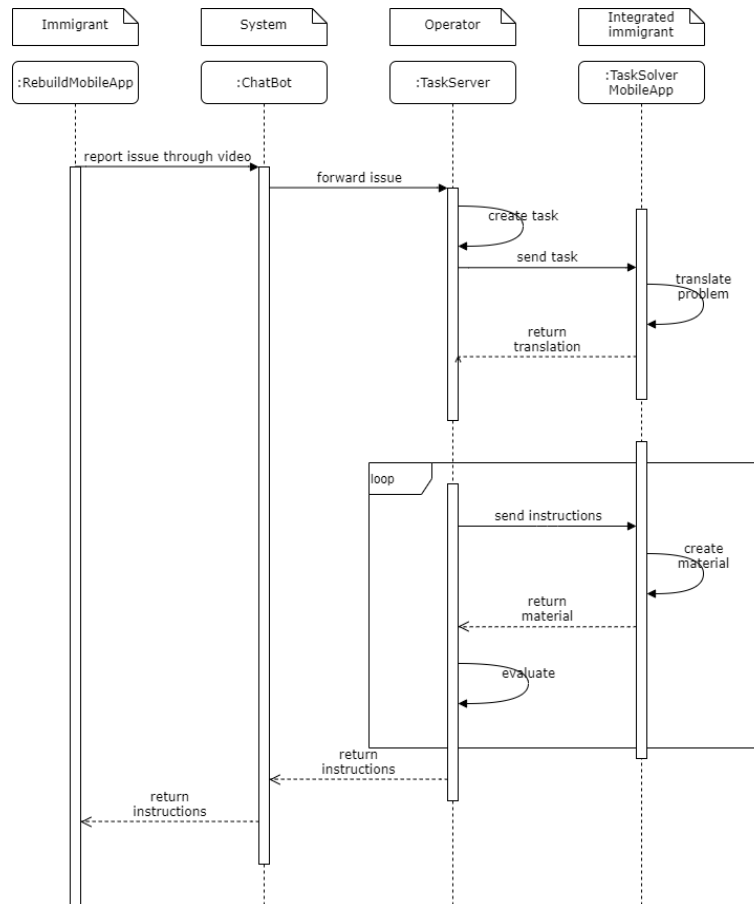
### 2.3.1 Request info from ChatBot using text

Responsible for the use case	CERTH
Use case pre-conditions	The user has to be registered to the ReBUILD App.
Use case post-conditions	The user will receive textual instructions on solving his/her issue.



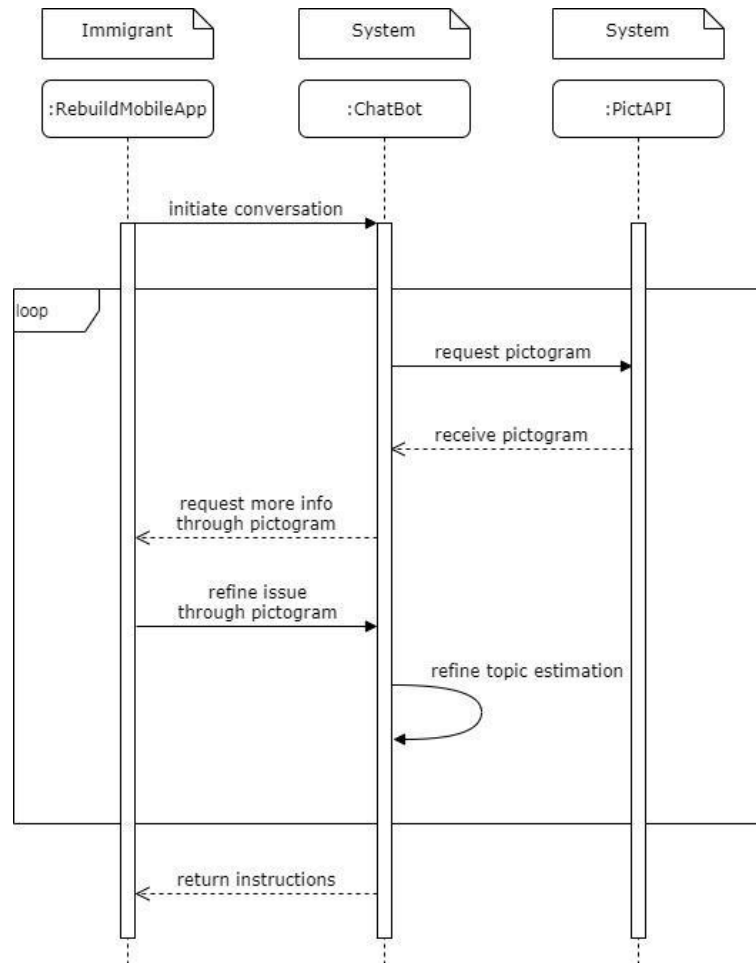
### 2.3.2 Request info from ChatBot using video

Responsible for the use case	CERTH
Use case pre-conditions	The user has to be registered to the ReBUILD App and be able to record short videos with his/her mobile device.
Use case post-conditions	The user will receive textual and audio-visual instructions on solving his/her issue.



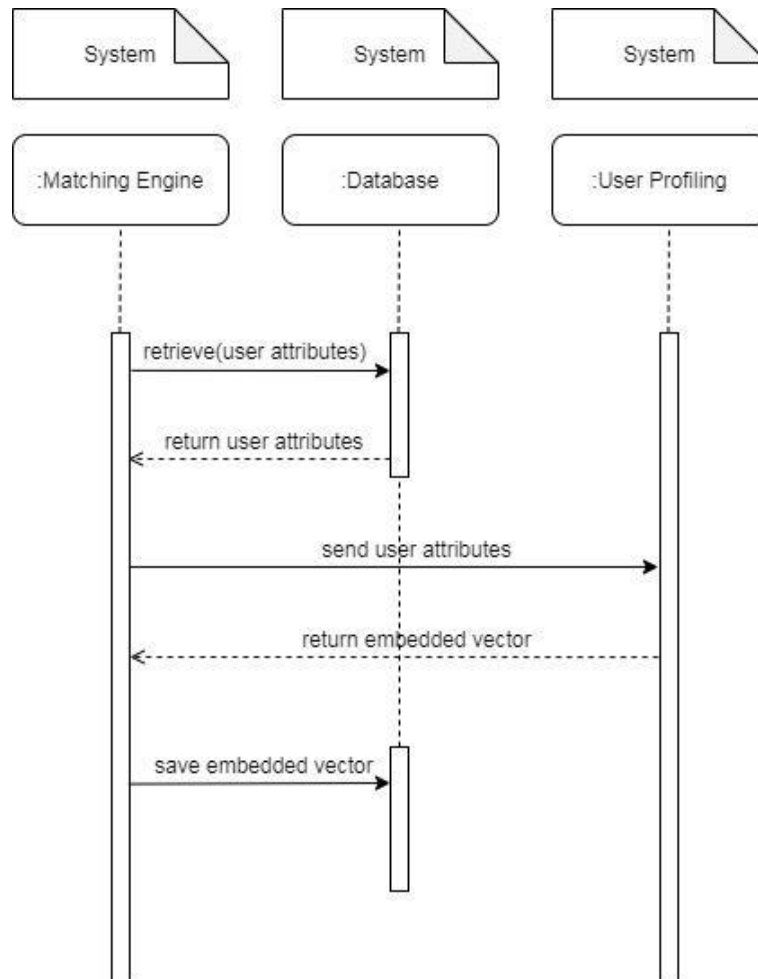
### 2.3.3 Communicate through Pictograms

Responsible for the use case	CERTH
Use case pre-conditions	The user has to be registered to the ReBUILD App.
Use case post-conditions	The user will navigate through pictograms in order to receive instructions on solving his/her issue



### 2.3.4 Build User Profile

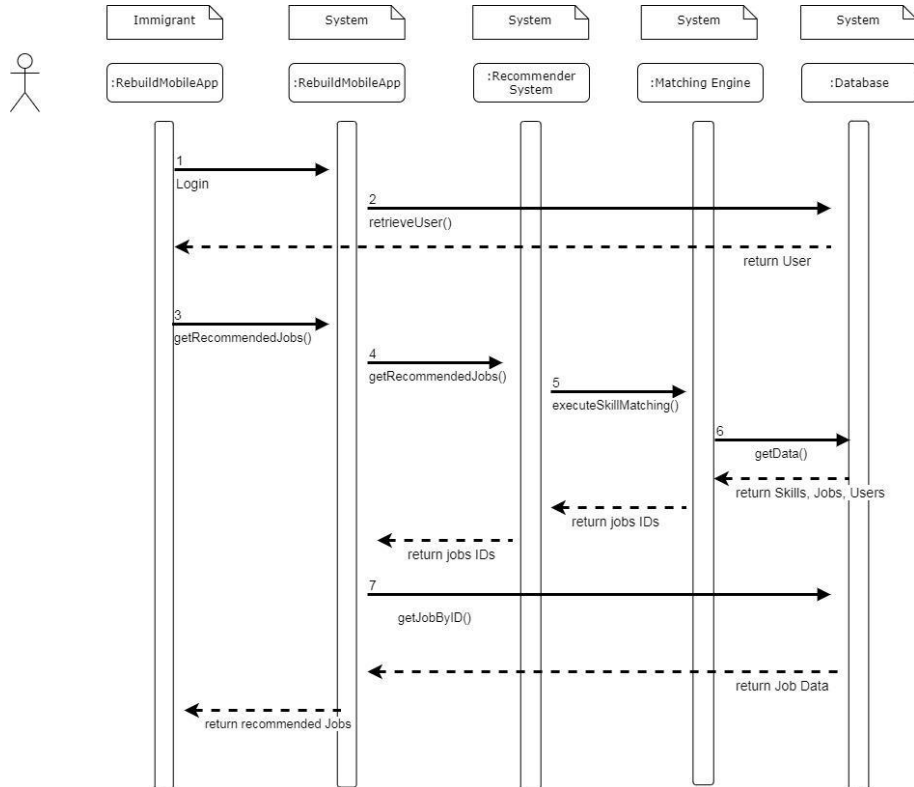
Responsible for the use case	CERTH
Use case pre-conditions	The user has to be registered to the ReBUILD App and fills in all the necessary forms regarding the matching cases.
Use case post-conditions	The matching component receives the embedded vector of the user profile.



### 2.3.5 Receive suggestion based on Matchmaking

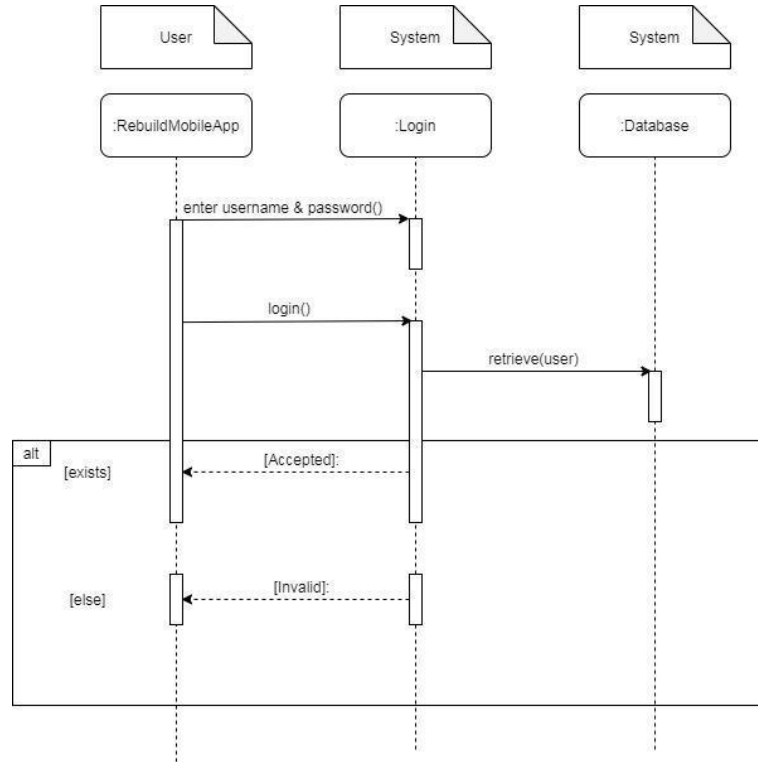
Responsible for the use case	UPM
Use case pre-conditions	The user has to be registered to the ReBUILD App and needs to provide some information (i.e. skills and hobbies for skill-based matching, and background for job seeking)
Use case post-conditions	The users will receive the list (ranking) of the job offers that fit the best for them





### 2.3.6 Authenticate

Responsible for the use case	ENG
Use case pre-conditions	The user has to be registered to the REBUILD App
Use case post-conditions	The user was authenticated and can surf through all the contents provided by the REBUILD App



### 3 INTEGRATION PLAN

This section aims at describing the **Integration Use Cases** and the **Infrastructure Requirements**.

The Integration Use Cases (Table 2) define how the components interact each other, what the platform behaviour is and also the effect after carrying out the interaction. In this way, it is possible to check if the platform is operating in the correct way or it needs to correct or improve some functionality.

The **deployment** of each component and the whole REBUILD Platform was executed through **Docker services**, where each component was deployed as a Docker container to make easier the deployment, installation and to optimize resource consumption of the whole platform.

REBUILD follows the **Security by Design** principle. All the components were deployed in a dedicated virtual machine in a network hosted in the server of ENG (Figure 3). Within the perimeter of the ENG network, the REBUILD components interact with any use of security token, since the component APIs were not public. Only the Data Base component was deployed in the UNINETTUNO server, even if the API layer to access to the DB was deployed within the platform perimeter. To guaranty the security of communications, all the interactions between REBUILD Data Base and REBUILD Data Base APIs were performed by using username and password and also other security communication protocol.

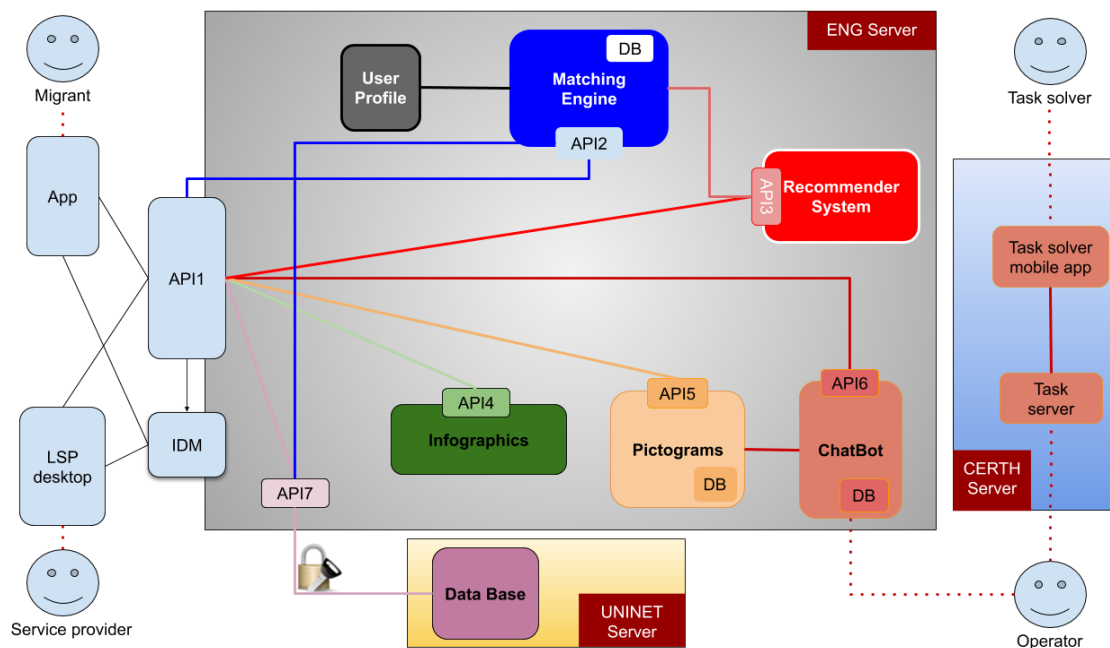


Figure 3: REBUILD Platform Deployment Diagram

Moreover, the Infrastructure Requirements (Table 3) and the installation guide of each component were described in Section 3.2.

To facilitate the reader below the components and its related short name are listed.

- DB: Data Base
- CB: ChatBot
- PicT: Pictograms
- RS: Recommendation System
- ME: Matchmaking Engine
- TS: Task Server
- TS\_MA: Task Solver Mobile App
- IS: Infographic Service



Below the definition of the column in the Integration Use Case table.

- Component
- Use Case
- Synopsys
- Involved components
- Dependencies from other Use Cases
- Prerequisites
- Behaviour
- Effect

The source code of all components is available in GitLab (<https://gitlab.com/rebuild-eu>).

## 3.1 INTEGRATION USE CASES

Table 2: Integration Use Cases

COMP.	UC	SYNOPSIS	INV. COMP.	DEPEND.	PREREQUISITES	BEHAVIOUR	EFFECT
<b>APP_UI</b>	APP_UI_01	Login	APP_UI, AAC		The user is already registered to the platform	Open the App homepage, press login button. A new page where AAC authentication takes place.	Upon window closure, the app is updated and user name is shown
	APP_UI_02	Registration	APP_UI, AAC	APP_UI_01	The user is not registered to the platform	Open the App homepage, click register button. A new page where AAC registration takes place.	The user is registered to the system and s/he can perform the login
<b>DB</b>	DB_01	Save user data	APP_UI, DB		The user is not registered in the APP	The user fills in the form to register	The user data are saved in DB
	DB_02	Save/Update user usage data	APP_UI, DB	APP_UI_01, DB_01	The user is registered in the system and logged in the APP	The user uses the APP and fills in the forms	The user data related to the usage of the APP are saved in DB
<b>CB</b>	CB_01	Communicate with ChatBot through text	CB, APP_UI, DB	APP_UI_01, DB_02	The immigrant is registered and logged in in the APP	The immigrant navigates through the ChatBot. The ChatBot uses text for the user navigation.	The immigrant expresses a need for information. The ChatBot covers this need.
	CB_02	Communicate with ChatBot through pictograms	CB, APP_UI, DB, PA	APP_UI, DB, PA	The immigrant is registered and logged in in the APP	The immigrant navigates through the ChatBot. The ChatBot uses text and pictograms for the user navigation.	The immigrant expresses a need for information. The ChatBot covers this need.
	CB_03	Communicate with ChatBot through video	CB, TS, MA_TS, APP_UI, DB	APP_UI_01, DB_02	The immigrant is registered and logged in in the APP. The integrated immigrant has a google account.	The immigrant navigates through the ChatBot. S/he does not manage to find the required information. S/he sends a video, which is send through the operator to the integrated immigrant. The operator and	The immigrant expresses a need for information. The integrated immigrant along with the operator cover this need.



						the integrated immigrant create a response with the required information.	
	CB_01	Communicate with ChatBot through text	CB, APP_UI, DB	APP_UI_01, DB_02	The immigrant is registered and logged in in the APP	The immigrant navigates through the ChatBot. The ChatBot uses text for the user navigation.	The immigrant expresses a need for information. The ChatBot covers this need.
<b>ME</b>	ME_01	Perform a matching of user profile properties with the LSP offer (i.e. type job or type mentor-mentee)	UP, ME, DB	DB_01, UP_01	The migrant fills the minimum information that the app asks	The immigrant clicks in the job seeking service, that will return a ranking with the most adequate jobs for him/her to apply based on his/her background	The outcomes are stored in the BD. These will be used by RS
	ME_02	Perform a query to extract the jobs /mentors available to perform an matching	ME, DB		Whenever an analysis is going to be performed, the module needs the information of the services offered and its status to evaluate whether these can be suggested to the migrant	The ME gets an API-KEY from the system to query the info to the DB	The DB returns the info to ME
<b>RS</b>	RS_01	Query the DB to check whether there is information relevant for the user	RE, DB		A task (b.b.d.) shall periodically (or under an event) to trigger this component	The module gets an APIKEY to query information relevant of the user or the LSP offer	This information is used to create recommendations that will be forwarded to the user
	RS_02	The RS performs an update of the events	RS, ME,UP,DB ,APP_UI	RS_01	This event is launch to look periodically, or under request the matching of mentor/or	Once the functionality is triggered, the system performs a task for the review of skills and job (calling the systems)	Recommendation are forwarded to the user via API



Re\_Build

		and recommendations of interest for the migrant			job that fits for the user, also access the Rebuild services offered (i.e. by LSP) to provide suggestions of interest to the migrant	and providing a set of periodical recommendations (i.e. new jobs that match your profile, new mentor, this event this week maybe of your interest)	
--	--	---	--	--	--	--	--

## 3.2 INFRASTRUCTURE REQUIREMENTS AND SETUP

The recommended environment for the REBUILD platform installation is described in the following table:

*Table 3: Infrastructure Requirements*

	Name	Version	Component
<b>Operating System</b>	Linux	>Ubuntu 18.04	DB-CB-TS-ME-RS-IS- PicT
	Android	9.0+ (API Level 28+)	TS_MA
<b>Execution Environment</b>	JVM	1.8+	All
	Python	2.7+	
	Node.js	10.x+	
	Apache Tomcat	8+	
	Npm	6.x+	
	Docker	18.06+	
	Docker-compose	1.23	
	Git	latest	
<b>Database</b>	Elasticsearch		ME-RS-IS-PicT
	MongoDB	3.6+	CB-TS-DB
	Google Firebase		TS_MA
<b>Storage</b>	80 GB		All
<b>RAM</b>	8 GB		
<b>Processors</b>	4 CPU		

### 3.2.1 Data Base installation and setup

This component based on MongoDB version 4.2. The CRUD APIs rely on spring-boot with java jdk8. To be compliant with the pseudonymisation the management of the persistence is based on spring-data-mongodb<sup>1</sup> and the encryption is demanded to spring-data-mongodb-encrypt<sup>2</sup>.

The latter component will be published as docker image into the Docker-Hub repository.

### 3.2.2 ChatBot installation and setup

This component is based on python, flask, chatterbot, network and elasticsearch. The initial version uses Facebook Messenger as a medium between the users (immigrants) and the rule-based engine or the operator. The final version will be included in the REBUILD app.

<sup>1</sup> <https://spring.io/projects/spring-data-mongodb>

<sup>2</sup> <https://github.com/bolcom/spring-data-mongodb-encrypt/>



It is embedded in a docker image. The docker image can be installed on any Ubuntu server with the Docker server package installed. No other software requirements need to be taken into account.

### 3.2.3 Task Solver installation and setup

This component is based on node js. It uses Google Firebase for authenticating, chatting and exchanging files with the Task Solver mobile app, so the operator needs to have a Google account.

It is embedded in a docker image. The docker image can be installed on any Ubuntu server with the Docker server package installed. No other software requirements need to be taken into account.

### 3.2.4 Task Solver Mobile App installation and setup

This component is a mobile application for Android phones (9.0+, API Level 28+). It is based on flutter and uses Google Firebase for authenticating, chatting and exchanging files with the Task Server, so its users, the integrated immigrants, need to have a Google account.

It can be installed through an ordinary apk package, which later can be delivered through Google Play.

### 3.2.5 User Profiling installation and setup

The component is based on python, tensorflow, flask and Elastic Search.

It is embedded in a docker image. The docker image can be installed on any Ubuntu server with the Docker server package installed. No other software requirements need to be taken into account.

### 3.2.6 Matchmaking Engine installation and setup

This component will be based on Neo4J a graph database and python for the matching. Additional libraries for python might be required (i.e. tensorflow or pytorch for deep learning) and flask for API development. Internal DB of neo4j will require a volume.

A docker image will be published in the Docker-Hub repository with is corresponding Docker file for replicability.

### 3.2.7 Recommendation System installation and setup

This component will be based Python and related libraries. For instance, some as tensorflow or pytorch for deep learning recommendation will be needed. Also, flask for API development. A docker image will be published in the Docker-Hub repository with is corresponding Docker file for replicability.

## 4 REBUILD DATA MODEL

---

The **REBUILD Data Model** was designed starting from the information coming from the use case scenarios. The Data Model maps all the information needed to develop the functionalities provided by the REBUILD App.

To be **compliant with GDPR**, all the data collected from the users will be **pseudonymised** and **encrypted**. When a user wants to take advantage of the Right to be forgotten, REBUILD platform will cancel each personal profile and personal data, keeping only anonymised data exclusively for statistical and research purpose.

Before collecting data from users, REBUILD will ask them to explicitly agree to an **informed consent** form where each service and related purpose of the data processing is described. Only once the user has signed (physically or electronically) the informed consent, REBUILD starts collecting and processing data. If the user does not agree with all (or some of the) terms of the informed consent, s/he will be informed that the services provided by REBUILD platform that require data the individual does not wish to provide cannot be delivered.

Even if the Data Base used in the REBUILD Architecture is a no E-R data base, the Figure 4 represents the E-R diagram of the REBUILD Data Model to better explain how the information is linked each other.

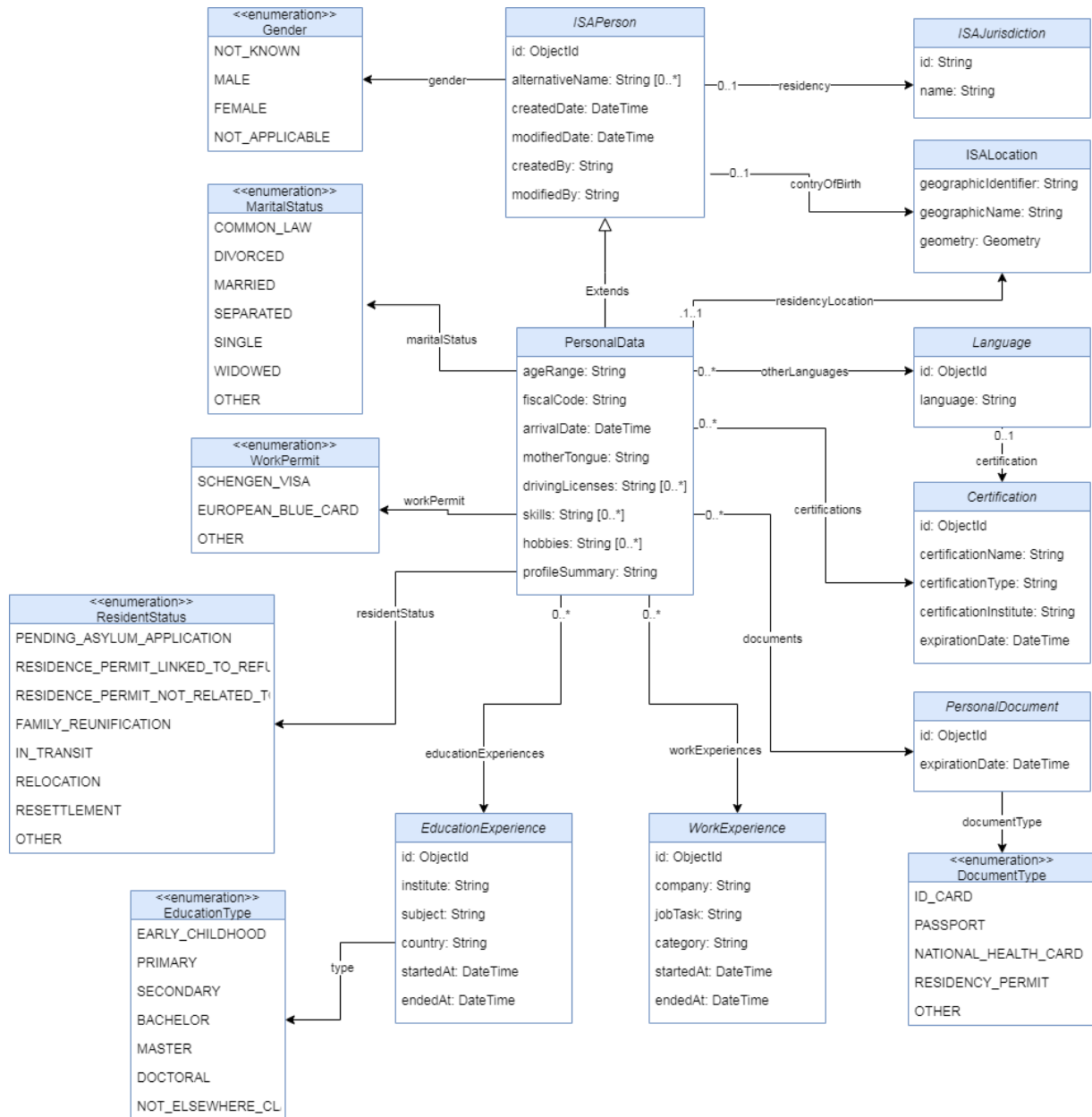


Figure 4: The REBUILD Data Model

The following sub-sections contain the Data Model split in general information and those ones coming from use cases. In order to make the REBUILD App interoperable, the Data Model is **compliant with the ISA<sup>2</sup>** (Interoperability solutions for public administrations, businesses and citizens) **Core Vocabularies** that are simplified, reusable, and extensible data models that capture the fundamental characteristics of an entity, such as a person or a public organisation, in a context-neutral manner [3].

## 4.1 ISA<sup>2</sup> VOCABULARIES

In this Section some information from ISA<sup>2</sup> vocabularies was included in the REBUILD Data Model.

*Table 4: ISA2 Vocabularies*

ISA Person			
Name	Property	Type	Description/Example
Alternative Name	alternativeName	String	Any name by which an individual is known other than their full name. This will be presented as nickname
Gender	gender	Code	[0=not known, 1=male, 2=female, 9=not applicable] ISO5218
Country of Birth	countryOfBirth	String	
Residency	residency	String	the region of the country where the user lives
ISA Legal Entity			
Legal Name	legalName	Text	The legal name of the business
Alternative Name	alternativeName	Text	
Company Type	companyType	Code	This property records the type of company
Company Activity	companyActivity	Code	The activity of a company
Identifier	identifier	Identifier	The identity relation links a resource to any formally issued identifier for that resource other than the one that confers legal status upon it
Registered Address	registeredAddress	Address	In almost all jurisdictions, legal entities must register a public address
Legal Entity	legalEntity	Legal Entity	This is useful, for example, where an organisation includes one or more legal entities
ISA Address			
Address Area	addressArea	String	the name or names of a geographic area or locality
Post Code	postCode	String	
Address Id	addressId	String	

## 4.2 REBUILD SPECIFIC DATA MODEL

The following table define the specific REBUILD Data Model, mainly coming from analysing the information coming from use case scenario

*Table 5: REBUILD specific Data Model*

REBUILD Personal Data - EXTENDS ISA Person			
Age range	ageRange	String	[18-24, 25-34, 35-44, 45-54, 55-64, 65-74, 75-84, 85-100]
Fiscal Code	fiscalcode	String	BDBDLN78E26D674V
Marital status	maritalStatus	String (ENUM)	[ COMMON_LAW, DIVORCED, MARRIED, SEPARATED, SINGLE, WIDOWED, OTHER;]
Arrival date	arrivalDate	Date	23/9/2015

Resident status	residentStatus	String (ENUM)	['Pending asylum application', 'Residence permit linked to refugee (or other protection) status', 'Residence permit not related to asylum', 'Family reunification', 'In transit', 'Relocation', 'Resettlement', 'Other']
Work permit	workPermit	String (ENUM)	['not desired', 'not yet in demand', 'pending', 'granted', 'free of work permit']
Mother tongue	motherTongue	String	French
Residency Location	residencyLocation	ISA-Location	
Other Languages	otherLanguages	Array<Language>	
Driving Licenses	drivingLicenses	Array<String>	['B', 'A']
Document	documents	Array<Personal Document>	
Education Experiences	educationExperiences	Array<EducationExperience>	
Work Experiences	workExperiences	Array<WorkExperience>	
Certifications	certifications	Array<Certification>	
Hobbies	hobbies	Array<String>	[golf, guitar, ...]
Soft Skills	softSkills	Array<String>	[teamwork, punctuality, ...]
Profile Summary	profileSummary	Text	Description of the profile
<b>REBUILD Certification</b>			
Certification Name	certificationName	String	
Certification Type	certificationType	String	
Certification Institute	certificationInstitute	String	
Expiration Date	expirationDate	Date	
<b>REBUILD Personal Document</b>			
Document Type	documentType	String (ENUM)	ID card
Expiration Date	expirationDate	Date	
<b>REBUILD Educational Experience</b>			
Type	type	String (ENUM)	[diploma, certificate, bachelor degree, master degree, PhD]
Institute	institute	String	University of Palermo
Subject	subject	String	Computer Engineering
Country	country	String	
Started At	startedAt	Date	Format mm/yyyy
EndendAt	endedAt	Date	Format mm/yyyy
<b>REBUILD Language</b>			
Language	language	String	Italian
Certification	certification	Certification	
<b>REBUILD Working Experience</b>			
Started At	startedAt	Date	Format mm/yyyy
EndendAt	endedAt	Date	Format mm/yyyy
Company	company	String	
Job task	jobTask	String	
Category	category	String	
<b>REBUILD Local Service Provider - EXTENDS ISA<sup>2</sup> Legal Entity</b>			
Brief description of service	serviceDescription	String	The service allows to book a medial appointment
Needed documentation	neededDocuments	Array<String (ENUM)>	identity card, social number, resident permit

Opening hours	openingHours	Array<String>	8.00-13.00
Website	url	String	ww.mxn.com
Email Address	emailAddress	String	wsk@uab.com
Telephone	telephone	String	+3925489543
<b>REBUILD Education</b>			
Title of the course	Title of the course	Title of the course	Title of the course
Description of the course	Description of the course	Description of the course	Description of the course
Organization	Organization	Organization	Organization
Prerequisites	Prerequisites	Prerequisites	Prerequisites
Link to the course	Link to the course	Link to the course	Link to the course
<b>REBUILD Mentor - EXTENDS Persona Data</b>			
Mentor status	mentorStatus	String (ENUM)	Student, teacher, PAS, Alumni (ex-students)
Availability	availabilities	Array<String>	Monday 13-14h / Tuesday 17-19h /...
Known location	knownLocation	ISA-Location	UAB Vila, UAB Campus, Sabadell (city names)
Language to speak	languageToSpeak	String	Spanish
Motivation to take part of the mentoring program	motivation	String	plain text
Contribution to the mentoring program	mentoringContributions	Array<String (ENUM)>	Showing the surroundings - Practing language - Creating new social links - Leisure time - Interpersonal competences - Others: fill it with your ideas
Expectation from the mentoring program	mentoringExpectation	String	plain text
Mentor challenges	mentorChallenges	Array<String>	
Matchmaking limitation	matchmakingLimitations	Array<String>	I am not comfortable pairing with a woman I am not comfortable pairing with a man
<b>REBUILD Mentoring</b>			
Mentor	mentor	Reference to Mentor	This is an identifier, the nickname of the user
Mentee	mentee	Reference to Mentee	This is an identifier, the nickname of the user
Meetings	meetings	Array<Meeting >	Meetings
<b>REBUILD Meeting Request</b>			
Mentoring	mentoring	Reference to Mentoring	
Date	date	Date	
Location	location	ISA-Location	
Mentee message	message	String	
<b>REBUILD Meeting Response</b>			
MenteeMeetingRequest	menteeMeetingRequest	Reference to the request	
Mentoring	mentoring	Reference to Mentoring	
Mentor acceptance	accepted	boolean	
Mentor message	message	String	
<b>REBUILD Meeting</b>			

Mentoring	mentoring	Reference to Mentoring	
Date	date	Date	
Location	location	ISA-Location	
Type of activity	activity	String (ENUM)	
Activity to increase mentee's autonomy	activityDescription	String	
<b>REBUILD Mentor Meeting Feedback</b>			
MeetingDiary	meeting	Ref Meeting	
Satisfaction	satisfaction	Number	1-5 range
Relationship improvement	relationshipImprovement	String	
Reluctancy	reluctancy	String	
Technical question	technicalQuestions	Array<String>	
Comment	comment	String	
<b>REBUILD Mentee Meeting Feedback</b>			
MeetingDiary	meeting	Ref Meeting	
Satisfaction	satisfaction	Number	1 - 5 range
Comment	comment	String	

## 5 REBUILD USER INTERFACES

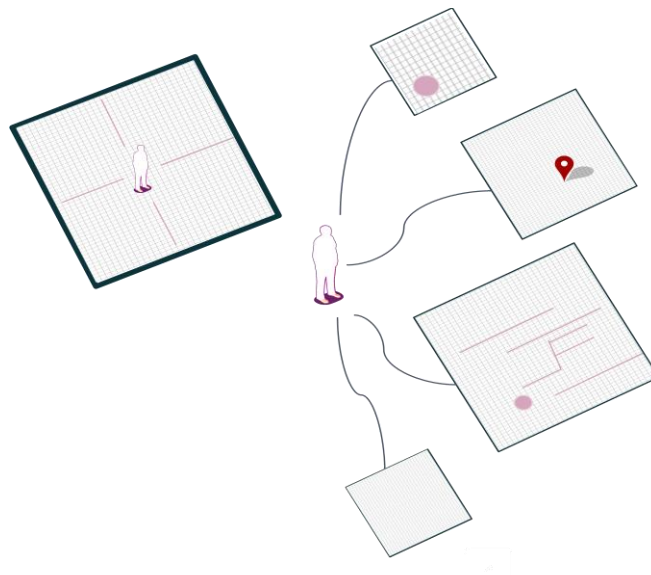
The aim of this section is to give an overview on the REBUILD user interfaces, focusing on the REBUILD Mobile Application and the idea on how to use it and what interaction model approach was defined to interact with it.

Main role of the REBUILD Mobile APP is to guide migrants both to explore the **private space and the neighbourhood** with intimate relations and local services to the person, and the **open and wide space**, with relations and connections to the home country and other places.

The app may foster the focus towards **close and local perspective** or enlarge the focus towards an **open and wide perspective** (Figure 5).

Different tiles bring the attention to separate map portions being:

- the vicinity
- the local city
- the host country
- the homeland
- other countries



*Figure 5: Private vs Wide approach*

### 5.1 USER INTERFACE MODEL

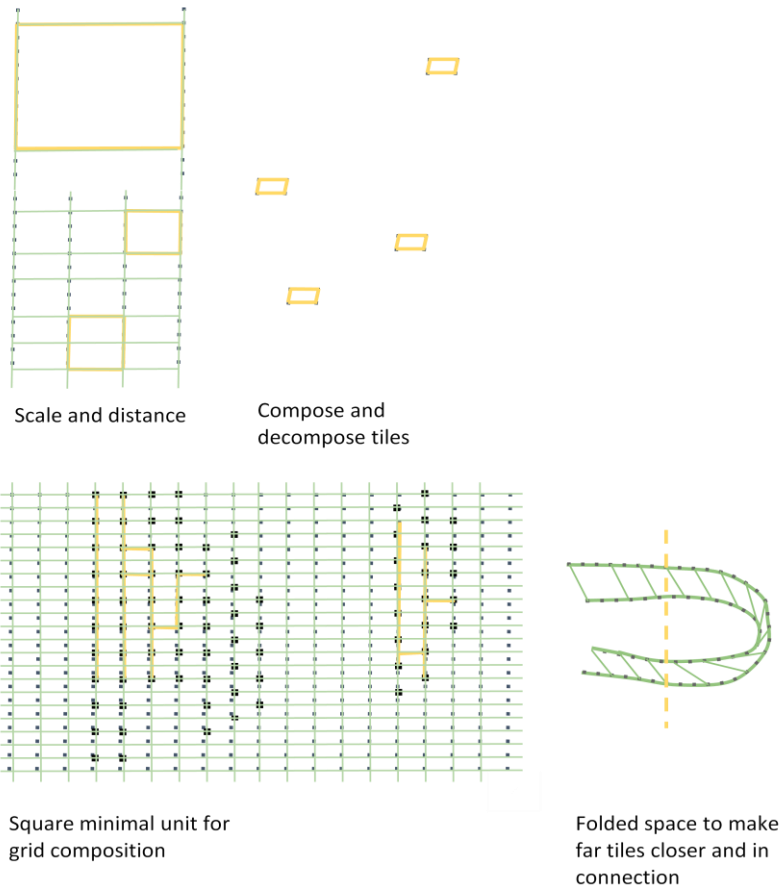
The user interface model adopted to develop the REBUILD mobile APP is based on the Dot grid model. The dot grid model allows the user interface to be dynamically shaped in squared tiles which may differ one each other by:

- position



Re\_Build

- dimension
- colour
- style



*Figure 6: Dot grid model*

The user interacts with the mobile app in graphical way

The dot grid model allows effectively managing different tiles bringing the attention to:

- places
- services
- people
- events
- connections

that may have attributes like:

- priority
- salience
- type

Suggestions based on

**Re\_Build**

- interests,
- needs,
- objectives.

According to criteria like

- priority levels
- background/ foreground

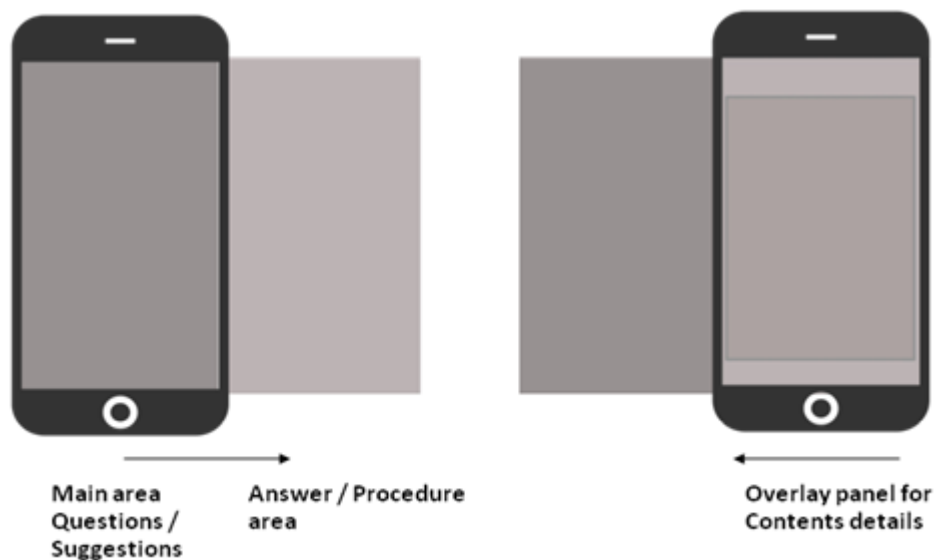
The REBUILD Chatbot provides suggestions, in a proactive way, according to user profiling and shows different and dynamic contents by differentiate:

- tiles' dimension
- tiles' position
- tiles' colour code

## 5.2 USER INTERFACE LAYOUT

The main layout of the REBUILD mobile APP is made up of three areas:

- MAIN: the first one dedicated to Bot suggestions and Questions building blocks
- PROCEDURE: the second one for the Answers provided by the chatbot
- CONTENT DETAILS: the third one for the content details as well as specific operations (like video recording)



*Figure 7: REBUILD mobile APP UI approach*

The main area (Figure 8) is divided in three sections:

- questions saved

- suggestions
- ask a question

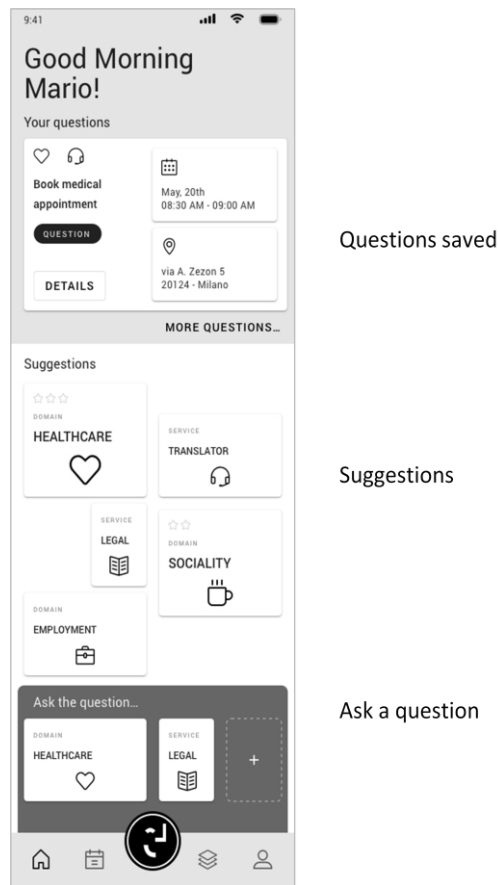


Figure 8: REBUILD mobile APP - main page

When a user asks a question (Figure 9), s/he has the possibility to write directly the question in the specific box on top of the page, or compose the question through the use of the pictograms.

If decide to use the pictograms, first s/he has to select the domain s/he is interested from the domain blocks. Then according to the selected domain the user has to select the needed service from the service block.

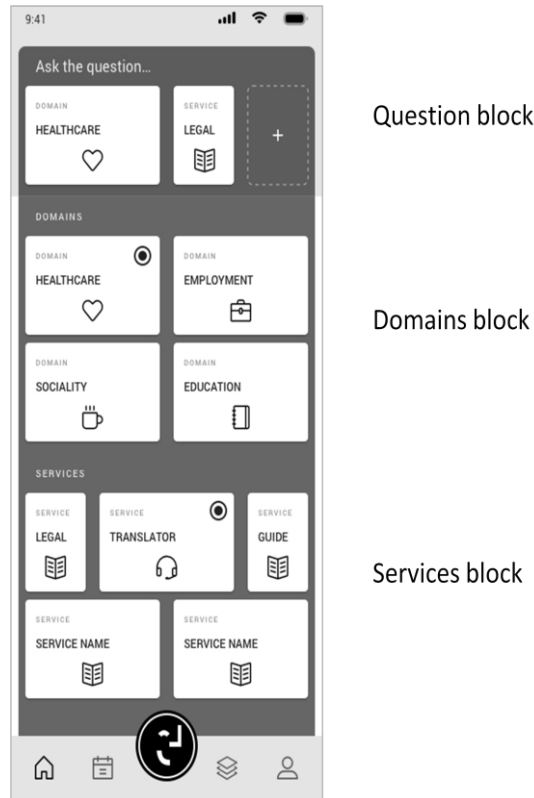


Figure 9: Ask a question through text or pictograms

Finally, after selecting the needed service the user can make the question making a video to be sent to the operators (e.g. Medical appointment Use Case Scenario [2])

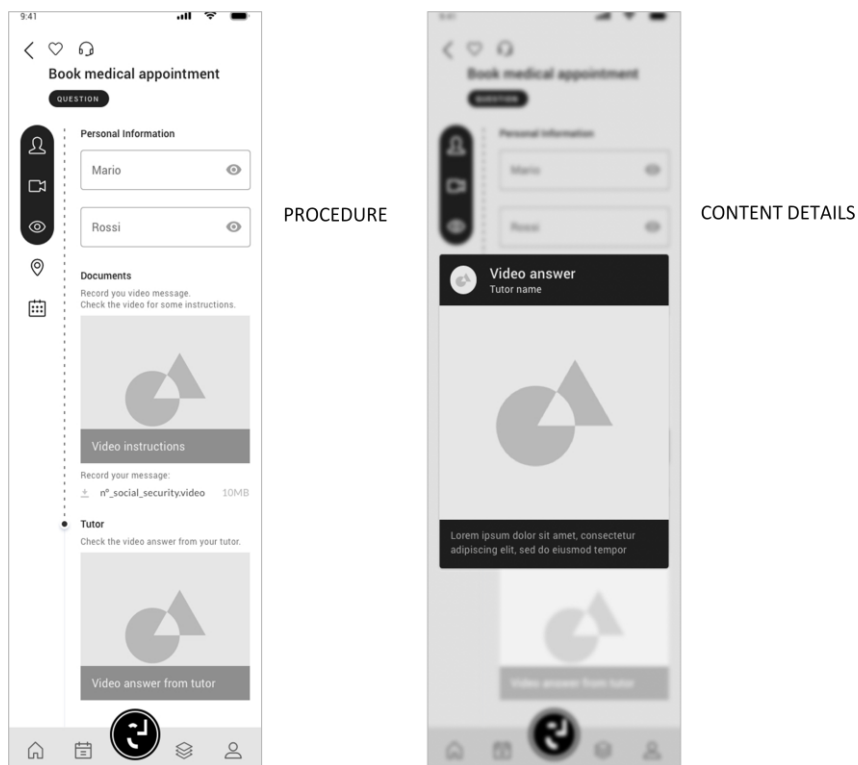


Figure 10: Make a question recording a video



## 6 CONCLUSION

---

At the moment of submission of this deliverable, the REBUILD technical team is working on the finalisation of platform integration. We are deploying and testing each interaction among components to be ready for the first internal testing phase (foresees in June 2020). From this first testing we expect to collect feedback from the users, mainly on the Data Base (to understand if all the information we are requesting now is enough or not, or we are requesting more information that we need to make the services operating).

In the Deliverable D5.3 (due to M36) we will describe all the updates and outcomes coming from the two phases of experimentation. We will take into account the feedback coming from the real use to adapt the REBUILD APP to their needs and in case make the appropriate changes to the Architecture and Platform.

## 7. REFERENCE

---

- [1] A. Filograna, «D5.1 - REBUILD Reference Architecture,» REBUILD project, 2019.
- [2] UNINETTUNO, «D2.5 - REBUILD Service Scenarios and Prototypes,» REBUILD project, 2019.
- [3] ISA2, «ISA2 Core Vocabularies,» [Online]. Available: [https://ec.europa.eu/isa2/solutions/core-vocabularies\\_en](https://ec.europa.eu/isa2/solutions/core-vocabularies_en). [Consultato il giorno 31 03 2020].



ICT-enabled  
integration facilitator  
and life rebuilding guidance

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215



## REBUILD

ICT-enabled integration facilitator and life rebuilding guidance

Deliverable: D5.2 Pilot Platform first prototype



This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215.