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
Project start date: 01/01/2019 | Duration: 36 months

Deliverable: D3.3 REBUILD recommendation environment and follow-up

DUE DATE OF THE DELIVERABLE: 30-03-2020
ACTUAL SUBMISSION DATE: 27-04-2020

Project	REBUILD – ICT-enabled integration facilitator and life rebuilding guidance
Call ID	H2020-SC6-MIGRATION-2018-2019-2020 – DT-MIGRATION-06-2018
Work Package	*WP3 – Data Analysis and skills matching*
Work Package Leader	*Universidad Politécnica De Madrid UPM*
Deliverable Leader	*Universidad Politécnica De Madrid UPM*
Deliverable coordinator	Gustavo Hernández ghp@gatv.ssr.upm.es
Deliverable Nature	Report
Dissemination level	Public (PU)
Version	1.0
Revision	Final

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

Document Info

Authors

Author name	Organization	E-Mail
David Martín Gutierrez	UPM	dmz@gatv.ssr.upm.es
Gustavo Hernández	UPM	ghp@gatv.ssr.upm.es

Document History

Version #	Author name	Date	Changes
0.1	David Martín	18-02-2020	Starting version
0.2	Gustavo Hernández	1-03-2020	Initial Contributions
0.3	David Martín	09-03-2020	Contributions in section 3
0.4	David Martín	12-03-2020	Extension of section 2
0.5	Thodoris Semertzidis	29-03-2020	General review
0.6	Annelore Hermann	21-04-2020	External peer review
0.7	David Martín	23-04-2020	Extension of section 2
0.8	Gustavo Hernández	24-04-2020	Conclusions, general review
0.9	Julia ZOMIGNANI BARBOZA	28-04-2020	Complete review
1.0	Gustavo Hernández	28-04-2020	Final Review

Document Data

Keywords	D3.3 REBUILD recommendation environment and follow-up
Editor Address data	Name: David Martín, Gustavo Hernández Partner: UPM Address: Av Ramiro de Maeztu 7 Phone: +34 91 464160 ext. 142 Email: [dmz,ghp]@gatv.ssr.upm.es
Delivery Date	27-04-2020
Peer Review	*thodoris semertzidis CERTH*

Acronyms

Acronym	Complete	Meaning
API	Application Programming Interface	A technological tools that permits the interconnectivity of multiple systems
VAE	Variational AutoEncoders	A Deep Learning Technique using for multiple tasks such as compression, classification and recommendation

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

Executive Summary

This deliverable is reporting the activities of REBUILD consortium to design and create a recommendation scenario based on the information of interest for migrants and aiming at guiding them through the resources available. T3.3 is the last task in WP3 (on top of the others), and is the one that provides the information that the user's will receive, therefore it involves all actors in the integration chain.

The scope of tasks in WP3 is totally interrelated, where the profiles embedding done in Task T3.1 (and reported in D3.1) is inserted in the T3.2 (Skills matching), to provide end-2-end matching (with an end ranking list) based on the inputs (generally for job seeking and mentor-mentee matching). This information is employed in the recommendation engine (T3.3) presented in this task. Furthermore, the information created in this module will also include connection to other services of interest.

The document is organised as follows: The first provides a literature review, selecting the most appropriate relevant recommendation approaches. It explores traditional approaches such as Content-based and hybrid approaches. This analysis allows us to understand what is the processing pipeline and also to use some existing ontologies (dapated to the REBUILD needs). Then, the second section is intended to describe the most relevant techniques, as well as the implementation details. The technologies employed for the development of this module are provided, as well as the details of the interaction and goals.

Finally, technical details of the physical deployment of this module, involving libraries and dockerization are provided. This deliverable has been submitted with a minor delay due to the unexpected situation derived from the COVID-19.

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

Table of Contents

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

Index of Figures

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

1 Introduction

This project follows a user-centered 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 2nd and 3rd 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.

More specifically, this document provides information of the work performed in WP3 for the creation of a recommendation system able to guide and to provide meaningful tips and suggestions to the migrants. The target subjects for recommendations were listed in WP2, and are more related to the job seeking as main concern for migrants, and the guidance on services available. Therefore, the recommendation engine presented here will mainly target those services.

Furthermore, this deliverable first will present the status of the art on Collaborative Filtering, Content based and Hybrid approaches to target this topic, while also Deep Learning approaches are presented. Then, the adoption of such subjects will be presented and the details on how to adapt the techniques to the contents and services available.

Finally, a set of technical details about the implementation of such services, how the ReBUILD recommendation engine will interact via API and the technologies are provided.

2 Recommender System: Background and State Of the Art

2.1 Introduction

The exponential growth of online multimedia content as well as many other sorts of information have fostered the incorporation of powerful recommender systems (RS) which can provide final users with specific and meaningful information in order to facilitate their lives in many aspects such as health, education, tourism, social entertainment etc.

Thus, as authors remark in (Osadchiy, T., Poliakov, et al 2019), a RS attempts to firstly identify what the consumer preferences are in a certain topic and subsequently, it suggests similar or relevant items that may be interesting for each consumer or user.

When talking about recommender systems, one may analyze different families and approaches that have been researched and implemented in the last years including Collaborative filtering or content-based recommendation procedures as it is described in (Pazzani, M. J., et al 2007).

Moreover, as in many other Machine or Deep Learning techniques, the goal of a RS consists in predicting new items using some data which in this case may correspond to the historical data of users or to the similarity among items.

2.1.1 The problem tackled

The main added value of the proposed Recommendation engine, is to tackle the identified need of guidance of ReBUILD users within the app. The ambition is to show in a personalised manner the services, and how these services can aid in their successful integration. In such context, taking into consideration the tools available in the ReBUILD ecosystem, the following support topics are covered:

Identified Need	Rebuild Support type
Job seeking	ReBUILD will periodically consult the Skill Matching service, to make recommendations based on the skill matching outcomes for the migrant. The recommendations will be provided under user subscription, and the user will have the decision to unsubscribe if desired.
Events of Interest	ReBUILD will check the offer of events (social, cultural ...) to suggest those ones of interest based on the user choices (preferences or topics of

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

	interest). Again, subscription to the service or external parties will be granted by the user.
Local Services Provider guidance	The recommendation will suggest the user with the most relevant services registered in ReBUILD. The user will have a search bar, where they can insert the topic of interest (i.e. legal: asylum seek), then the system will guide him/her automatically to the verified LSP that will offer the specific service related.
Perception and data spotting	The recommendation system will permit users to search information and make contrast again reliable information on a topic. The perception module will be directly connected to the recommendation to query the appropriate results.

2.2 Collaborative Filtering

A Collaborative filtering can be defined as a recommender system which provides recommendations based on the user preference models which can be created from either explicit or implicit features and measurements from the interests that a particular user may have (Osadchiy, T., Poliakov, et al 2019).

Therefore, Collaborative filtering only requires the historical user preferences and interests on a set of items in order to be implemented and designed. Additionally, when we mentioned explicit metrics or indicators, we referred to "making users assign" some ratings to the items. Furthermore, implicit metrics are more complex since they try to measure the user preference indirectly by for instance considering how much time a certain user spends looking at a particular item, the total number of clicks they did in a certain page of a website etc.

Moreover, the general idea behind Collaborative filtering is the following: users with similar preference models in the past will maintain similar tastes in a near future. Then, considering users or consumers with similar preferences and interests that rated in a positive manner a set of items, then, we can recommend these items to new target users who seem to have similarities with the former group of consumers.

When talking about similarity between a pair of users, we refer to correlations when analysing their purchasing history as well as ratings among the different items. Therefore, it is clear that this modality of the recommender system is strongly dependent on the amount of both historical and profile information of users. As a result, this approach will be more effective and robust whenever there is a considerable number of user preferences.

There are many approaches to address the Collaborative Filtering method. The simplest one is based on the so-called Nearest Neighborhood algorithm. Two main situations can be remarked: User-based and Item-based. The former consists on the following idea: having a certain (NxM) matrix of ratings, a set of users $u_i,\ i = 1, \ldots, n$ and a collection of items $p_j\ j = 1, \ldots, m$, the goal consists in predicting the

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

rating r_{ij} in case a target user u_i did not rate an item p_j. Thus, we need to determine a certain similarity between our target user and the rest of users, retrieve which are the top K similar consumers and then take some statistics (weighted average) regarding ratings from these K users with similarities denoted as weights. Moreover, there are different ways of computing this similarity including Pearson Correlation or the so-called cosine similarity. On the other hand, Item-based filtering relies on the hypothesis that two particular items are similar whether they receive alike ratings from the same user. Thus, the process consists in predicting for a given target user on an item using the weighted average of ratings on the most K similar items or elements from this user. The main limitation of the Nearest Neighborhood algorithm is that it cannot manage sparsity properly and is a common scenario in recommender systems applications.

Another method to be employed in this scenario is matrix factorization where the main goal is the decomposition of the original sparse matrix into a lower-dimensional matrix where latent features are included and the level of sparsity is considerably reduced. In fact, the matrix factorization provides a measure of how much a certain user is aligned with a set of latent factors and how much a particular item first into this particular set of latent features (Shuyu Luo et al. 2018).

However, the aforementioned techniques are limited to linear transformations which are not always the more adequate assumption in real scenarios and thus, many researchers started several investigations focussed on Deep Learning techniques in order to mitigate and improve the performance of the system in sparsity scenarios.

2.3 Content-based Filtering

The second strategy when talking about recommender systems is the so-called content-based technique (Pazzani, M. J., et al. 2007). In this case, the system analyzes the item content itself instead of the user preferences based on ratings. Therefore, the goal is to incorporate some tags or metadata to the set of items or products and then use this collection in order to recommend similar products based on this criterion.

Hence, content-based filtering considers recommending items to target users based on the ratings provided by them in similar items in the past (Amato, F., Moscato, V., et al. 2019). However, there is a crucial limitation to this kind of filtering - the so-called overspecialization which basically implies the recommendation of just similar items to those already rated by the user. Moreover, this approach needs the definition of an effective criteria to measure similarity between a pair of items based on some content-features which can be computationally expensive when talking about multimedia information such as video, images and audio. In these cases, these sorts of data are analysed using artificial intelligence techniques in order to extract powerful feature vectors that better characterize them and which can be incorporated into a similarity matrix.

2.4 Cold-start Problem and Hybrid Approaches

We have analysed that both Content-based and Collaborative Filtering approaches require a considerable volume of information regarding consumers including both user preferences and profiles as well as items.

Thus, one of the main limitations in this sense is the existence of sparse data sets or poor user profiles since it will drastically deteriorate the performance of the recommendations. This scenario is the so-called cold-start problem, where the users available in the system have limited information (Shaw, G., Xu, Y., et al. 2010). The same scenario is given when the volume of items is very large and users have only rated a few samples (Pajuelo-Holguera, F., Gómez-Pulido, et al. 2020).

Consequently, many researches have focussed on how the aforementioned scenario can be mitigated. Most of them have investigated Hybrid approaches (Schein, A. I., Popescul, et al. 2002) which combine collaborative and content-based filtering to improve the initial results of the system. Moreover, other researchers follow different strategies such as analysing user preferences based on demographic parameters (Lika, B., Kolomvatsos, K., et al. 2014) or using what is denoted as Social knowledge-based approaches such as the one presented in (Carrer-Neto, W., Hernández-Alcaraz, et al. 2012), (García-Sánchez, F., Colomo-Palacios, et al. 2020). Other authors are focussed on different approaches such as associated rules (Osadchiy, T., Poliakov, et al. 2019).

2.5 Deep Learning Approaches in Recommender Systems

Furthemore, the emergence of Deep Learning (DL) techniques has fostered applied research in recommender systems where they have been demonstrated to outperform previous solutions in many field domains. Moreover, DL models have the intrinsic nature to do all-purpose parameter fitting using advanced stochastic gradient descent procedures while achieving both speed and scalability throughout the use of Graphical Processing Units as authors remark in (Kiran, R., Kumar, P., et al. 2020).

However, deep neural networks require large volumes of data to be properly trained and evaluated, thus, it can only be used either when the amount of complexity of the scenario cannot be solved via previous traditional methods or when the amount of information is considerably large to train deep neural networks in an adequate manner.

As authors mention in (Zhang, S., Yao, L., et al. 2019), there are some considerations regarding deep neural networks that may have to be taken into account when using them in recommender system applications such as:

- Nonlinear transformations, capturing more complex and hidden user/item patterns contrary to previous methods such as matrix factorization or sparse linear models which can only work in linear assumptions.

- Embedding representations which are compact representations of the input due to the effectiveness of deep neural networks in learning the underlying latent factors of the collection of inputs.
- Sequential modelling which can be used to find sequential structures in data including mining the temporal analysis of user behaviour as well as the evolution of some items when talking about recommender system cases.
- Flexibility which implies the modular development of these models using different well-known frameworks such as Tensorflow or Pytorch. Therefore, they can be used to build powerful recommender systems.

Moreover, there exist some categories in order to organise these approaches but mainly they can be divided into two general ones: recommendation with Neural Building Blocks and Recommendation with Deep Hybrid Models. The latter category can be also separated into eight subcategories according to the deep learning model employed on each ase: Multi Layer Perceptron (MLP) (Lukasiewicz, T., Miao, Y., et al. 2017), AutoEncoder (AE) (Zhang, S., Yao, L., & et al. 2017), Convolutional Neural Network (CNN) (Yu, W., Zhang, H., et al. 2018), Recurrent Neural Network (RNN) (Xie, R., Liu, Z., et al. 2016), Restrictive Bolzano Machine (RBM) (Xie, W., Ouyang, Y., et al. 2016), Neural Autoregressive Deep Approaches (NADA) (Zheng, Y., Tang, B., et al. 2016), Neural Attention (NA) (Wang, X., Yu, L., et al. 2017), Adversary Networks (AN) (He, X., He, Z., et al. 2018) and Deep Reinforcement Learning (DRL)-based systems (Choi, S., Ha, H., et al. 2018). The latter branch is composed by hybrid models (Zhang, F., Yuan, et al. 2016), (Gao, J., Pantel, P., et al. 2019) which utilize a set of deep learning techniques in order to accomplish the recommendation by making use of the flexibility of neural networks.

3 Recommender System Within REBUILD Project

The scope of this section is the description of the recommender system which is included within the REBUILD project in order to facilitate migrants with both their integration and first contact in the arrival country. Difficulties in both communication and social habits in some typical life aspects such as healthcare, financial issues or leisure activities are considered to be mitigated by incorporating a recommender system which supports migrants in some of these situations.

To assess this objective, the system must analyse the different user profiles of the migrants in order to find similarities and dissimilarities among them throughout clustering procedures. Then, based on the user preferences in terms of services such as job seeking, housing or educational content, the system will provide new target users with notifications and recommendations of services or events according to what previous similar users rated.

Consequently, the more adequate approach to follow within the REBUILD framework is indeed an hybrid recommender system which considers a collaboration among users (Collaborative filtering approach) throughout a social knowledge-based strategy as well as an interaction among the elements that can be recommended in the platform (services, multimedia material etc.) based on their

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

similarity (Content-based approach). Thus, by combining these two techniques, some of the classical drawbacks of recommender systems as the one denoted as cold-start problem, can be mitigated in this application.

This section is also devoted to presenting the general architecture of this module, including its relationships with the rest of the components of the project such as the user interface application, the skill-matching or the user profiling modules which are also part of the analytics section of the project. Then, we will describe the main inputs of the system as well as the target outputs that must be considered. Subsequently, the different techniques and algorithms suggested in this application will be presented. Finally, some of the main use cases considered in this preliminary version of the whole system will be introduced in order to understand how these scenarios may support and facilitate some of the main issues that migrants have to face once they arrive in a new country.

3.1 System Overview, Architecture and Design

The recommender system is going to provide recommendations and notifications to migrants and other end-users from the following set of topics:

- Job opportunities based on their skills and competences.
- Educational and instructive contents such as language course or any other available content in the platform.
- Social events around the location of users to encourage them to get to know the culture and customs of the arrival country.
- Social mentoring which is one of the use cases considered within the project where new students enroll this service to receive some advice and support from mentees which are assigned to them according to the kind of profile that users (students in this case) prefer.
- Relevant locations and places for Healthcare and social assistance in order to support migrants in basic life aspects.
- Additional digital content material available in the platform such as videos, documents or any other sort of information to smoothen their integration in a new country.

In Figure 1, a general architecture of the REBUILD platform is presented. Within the back-end of the application, there is a set of components which are interconnected throughout APIs to make them available for internal communication among the components.

In particular, the recommender system is directly connected to all the rest of the components from the WP3 including both the user profiling and the skill-matching. The communication between these components is managed throughout an API whose initial development will be shown in a further section.

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

As expected, the recommender system needs to have access to the complete data stored in the platform in the data model schema that was previously defined within the REBUILD project. This includes having access to Users, Jobs, Skills, and Items such as multimedia content, other services as well as a range of further information available in the platform.

Consequently, a Latent Factor Model will be built where Users and Items will be connected throughout a set of latent features. This knowledge-based structure is illustrated in Figure 2 where the users are represented in the left side in the green rectangle, the collection of items are drawn inside of the green rectangle and finally, the latent space is represented as a purple rectangle. Therefore, the main objective in this scenario is the calculation of the latent space of features that connect both users and items in a more compact manner.

Then, this information is mapped into a graph representation using a powerful framework named Neo4j[1] which can be easily integrated in web applications via Docker containers. More specifically, this part of the analysis is provided in the skill-matching component as it is described in Deliverable 3.2. Moreover both users and items are connected based on the relationships that were also included in the preliminary data model of the REBUILD platform.

[1] https://neo4j.com/

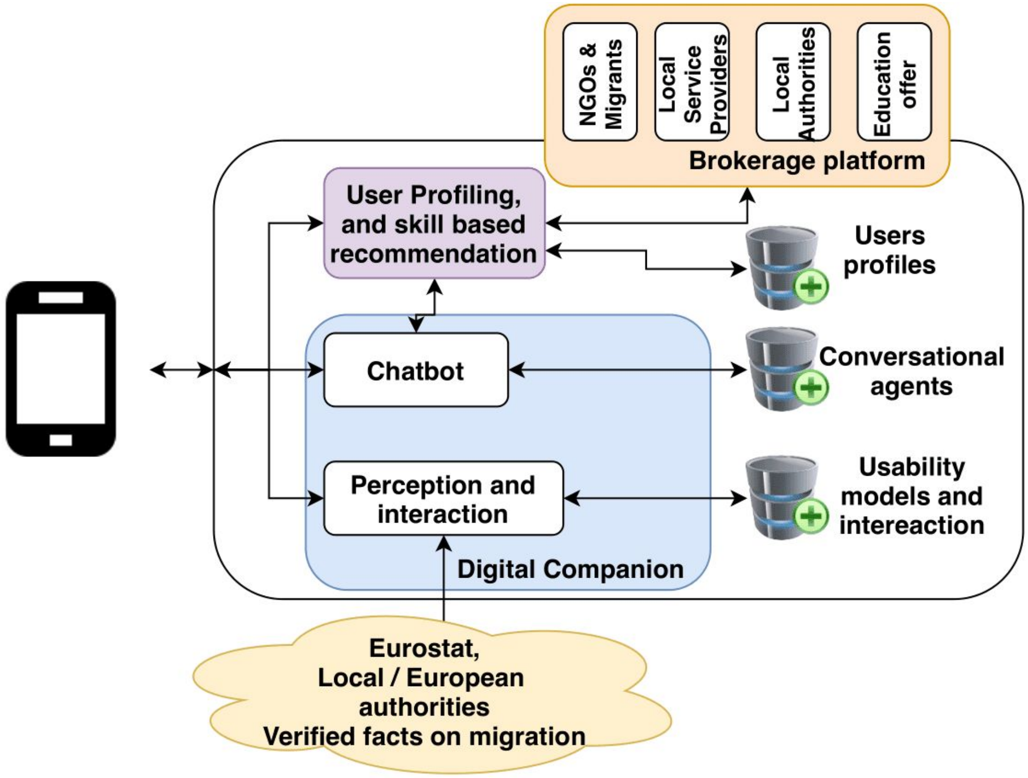

Figure 1: General architecture of recommendation engine within the REBUILD platform.

Furthermore, the recommender system component contains its own Application Programme Interface (API) which has different endpoints depending on the kind of recommendation that is meant to be triggered. Since the REBUILD project is a user-driven application, some of the functionalities may undergo modifications in future release according to the decisions made during the co-creation workshops as well as the GDPR terms which may provoke a slight variation of some of the elements involved in the data model.

In addition, regarding the integration of the component within the REBUILD system, the recommender system can be seen as a modular component, since it is dockerized into a virtual container, to be very easy-to-integrate. Moreover, it is developed in pure Python code using a well-known Web Framework named as Flask[2] which will be the back-end of the component.

Finally, this component will have its own Database in order to speed up some batch processes such as the incorporation of data to the recommender system, the incorporation of such information to the NEO4J graph or the calculation of similarity metrics which are needed for the generation of recommendations.

[2] https://flask.palletsprojects.com/en/1.1.x/

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

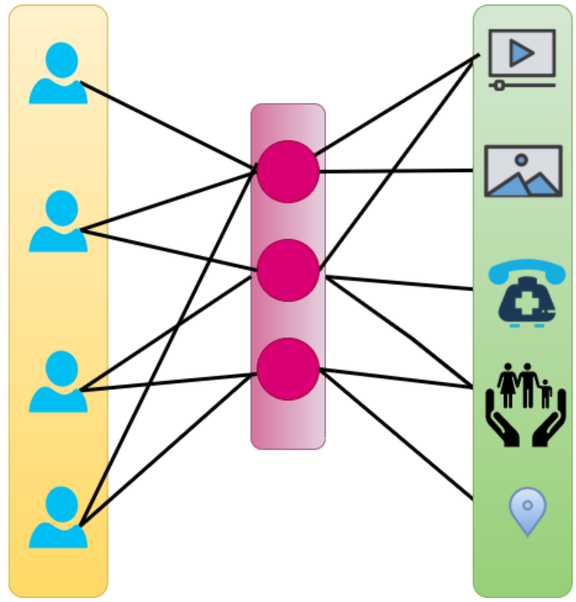

Figure 2: Illustrative representation of a Latent Factor model, where users are represented in the yellow rectangle, items are represented in green and finally, the latent features are within the purple one.

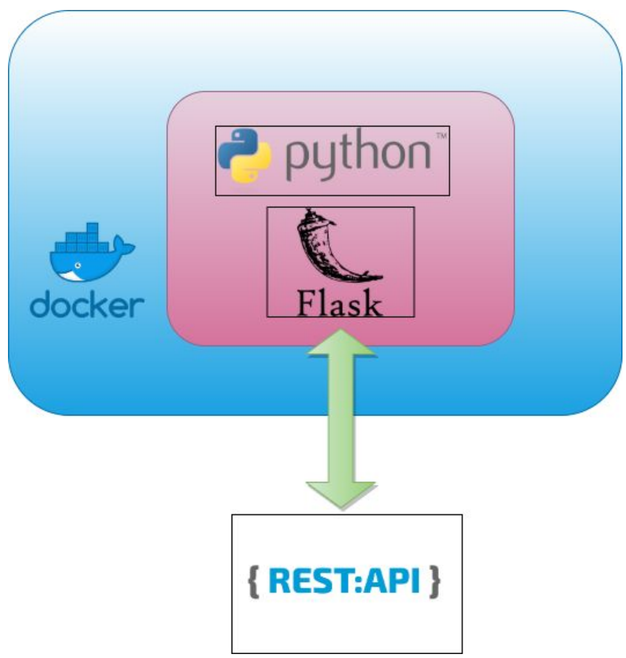

Figure 3: Graphical representation of how the recommender system component is implemented to be integrated as a flexible module within the REBUILD platform.

3.2 Recommender System Algorithms

The goal of this section is to present and describe the main algorithms that are performed to determine the recommendations that are more appropriate or adequate regarding the user needs. It is important to remark that the performance of the system depends directly on the amount of data available in the database of the project in order to avoid the classical problems that a recommender system faces such as the ones commented in previous sections including the cold-start or the overspecialization issues.

Moreover, the recommender system within the REBUILD project is based on a **Hybrid filtering** in order to improve not only the profiling of users but also the similarity among different contents or information (items) which are available in the REBUILD platform and thus, which can be provided to final users via the main REBUILD application.

In the following sections, we will present different stages and approaches that are considered when building the recommender system. Most of them are data-dependent so that several experiments will be needed to run once the whole data from the pilots available within the REBUILD application in order to decide which model works better in our scenario.

3.2.1 Knowledge Graph

The first step consists in incorporating the different elements of the database including User and Items into a graph database via Neo4j. We will denote this database as $G(E, V)$ which represents the edges or nodes as well as the relationships or vertices of the Graph. As expected, the vertices which are considered in the analysis, are the ones defined in the global Data Model of the REBUILD project.

The advantage of generating a social knowledge graph considering all the elements involved in the processing including Users and Items (Jobs, Skills, Events, Digital Content) lies in the fact that we can easily analyse how the items are connected to find potential similarities among them (content-based approaches) as well as how the users behave in terms of which items were associated to them in the past (collaborative-filtering). Moreover, the recommender system will receive as input the embeddings of the available users so that one can estimate the similarity matrix from them to be incorporated into the social knowledge graph when comparing different users in order to query similar candidates to a given target user.

Additionally, using a knowledge graph is really helpful in order to provide explainability of the results as well as to deal with data sparsity which is a potential issue to be considered in any recommender system scenario.

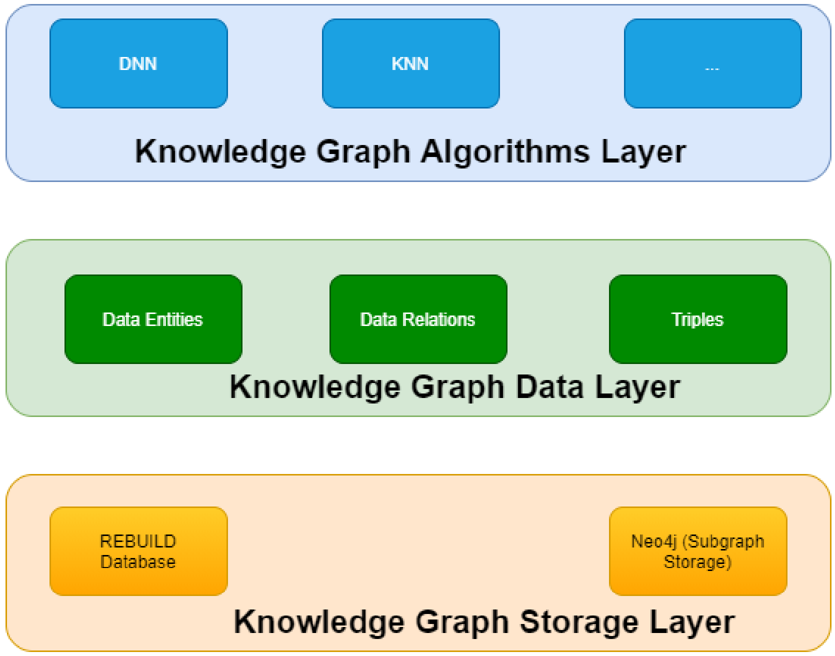

Figure 4: Graphical representation of the different layers that are needed to build the recommender system using an Hybrid approach that considers Knowledge Graph. Inspired by (Xianxian H. et al. 2019).

In Figure 4, the distinct stages that are considered for building the proposed recommender system are depicted. More specifically, the first layer consists in storing the data in both REBUILD database as well as in Neo4j. Then, the Data Layer is responsible for creating the entities, relations and triple. Basically a triple can be defined as follows:

$triple = (e_i, r_k, e_j)$, where $e_k, r_k \in R$. Therefore, e_i, e_j are entities, E is a entity set, r_k is a relation and finally R is the relation set. $triple = (e_i, r_k, e_j)$, where $e_k, r_k \in R$. Therefore, e_i, e_j are entities, E is a entity set, r_k is a relation and finally R is the relation set.

Finally, the last layer is responsible for computing the different algorithms to retrieve the best items to be recommended to each user, as well as to compute similarity among users and items. As we have commented before, when we talk about entities we refer to the complete collection of Users and Items (all the types) that are included within the REBUILD data model.

Consequently, using a social knowledge graph to incorporate insights to the recommender system is a crucial step to extract relevant information from the distinct entities that are involved in the system. To do so, Neo4j is employed as the graph framework in order to compute the operations that are

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

needed to add, update and delete data as well as to build powerful graph queries to retrieve the needed information.

3.2.2 Knowledge Graph Embeddings

There exist many approaches to generate the embeddings that will be used as input to feed the learning-based algorithms. Inspired by authors in (Wang, H., Zhang, et al. 2018), there are different approaches to build the embeddings:

- **TransE** which is a classical embedding procedure to be used when handling data with multiple relations. Denoting (h,r,t) to represent a particular triple, the goal of this method consists in the approximation of the target entity as the sum of the relationships and the input entity, so that, $t \approx h + r$.
- **TransH** allows the collection of entities to have distinct manners or representations when they are involved in different relations via projecting the entity embedding into a relation hyperplane so that $\dot{t} \approx \dot{h} + r$.
- **TransR**, which is very similar to **TransE** but in this case, we attempt to build the relation in different spaces for different relation types so that, by transforming in using a projection matrix M_r.

All the aforementioned procedures optimize the following margin-based ranking loss when training the models:

$$L = \sum_{(h,r,t) \in \Delta} \sum_{h',r,t' \in \Delta'} max(0, f_r(h,t) + \gamma - f_r(h',t'))$$

Where γ is denoted as the margin term and $f_r(.)$ is the scoring function and Δ, Δ' are the sets of correct and incorrect triples respectively.

3.2.3 Learning-Based Models

One of the main goals of using Machine Learning (ML) or Deep Learning (DL) approaches on top of the Knowledge Graph is to automatically extract hidden patterns as well as more powerful representations from the input data.

Moreover, in this case, the system would need to predict new vertices or relationships when new Items or Users are added in the database based on similarity metrics and embedding representations.

3.2.3.1 Autoencoders

One of the approaches that can be followed to make this kind of predictions are the so-called Autoencoders (AE) or Deep autoencoders (DAE) which attempt to find a low-dimensional space of the raw input data that represent it better in a more compact way. More specifically, an AE contains the following structure:

- Input Layer
- Hidden Layer(s)
- Output Layer

The most peculiar aspect of this model is that both input and output layers are the same. Moreover, the hidden layer is generally called encoding since it has a lower number of neurons in comparison with both input and output layers. Therefore, the idea is to reconstruct the raw input data using a lower representation of it. To do so, we need to force the system to learn hidden patterns from data as well as correlations using a smaller number of neurons in comparison with the input. The transition between the input and the hidden layer is called **Encoder** whereas the **Decoder** is referred to as the transition between the hidden layer and the output.

Autoencoders are very useful to find the latent feature representation between Users and Items as it is shown in Figure 2.

3.2.3.2 Neural Matrix Factorization

It is a new framework which attempts to generalize the so-called Non-Matrix Factorization problem. In particular, the model takes as inputs both the user embedding (user profiling provided as input of the system in REBUILD) as well as the Item embedding. On the other hand, the output consists in a number between 0 and 1 and represents the probability that a certain user may be interested in a certain item.

3.2.3.3 Restricted Boltzmann Machine

Another approach to deal with the problem consists in using Restricted Boltzmann machines which basically are denoted as generative stochastic neural networks and contain only an input layer and a hidden layer which is used to learn a certain probability distribution over the vector of inputs. This approach can be very useful to associate new Items with tags using this model.

Moreover, it can be used to recommend available items to new candidates or Users. As an example, considering a new candidate that has a set of skills, we can recommend the most appropriate jobs using this approach.

One of the main advantages when employing this model is the non-linearity transformation that it provides to find hidden patterns as well as the high level of interpretability that it has. In addition, this method is complex when speaking about training as authors remark in (Quentin Bacuet , 2019).

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

3.2.3.4 Deep Neural Network

Another approach is the use of classical Deep Neural Networks (DNN) taking as input the embedding vector of the user and having as output the most suitable set of recommendations for the input user. As expected, the output has the size of the number of Items available in the dataset.

Furthermore, the main advantage of using this model to perform the recommendations is that it provides non-linear transformations and also a fast query time once the model is trained properly. However, the main drawback is the lack of interpretability since the model is able to detect automatically hidden patterns from the input data.

3.2.3.5 Variational Autoencoder

A Variational Autoencoder (VAE) has the peculiarity of incorporating a sampling layer instead of a simple dense layer in the encoding layer as a classical autoencoder. More specifically this sampling layer will use both the mean and variance from the last layer of the encoder in order to create a Gaussian sample. Then, it is passed through the decoder to generate an output. In the following figure, the architecture of a typical VAE is shown.

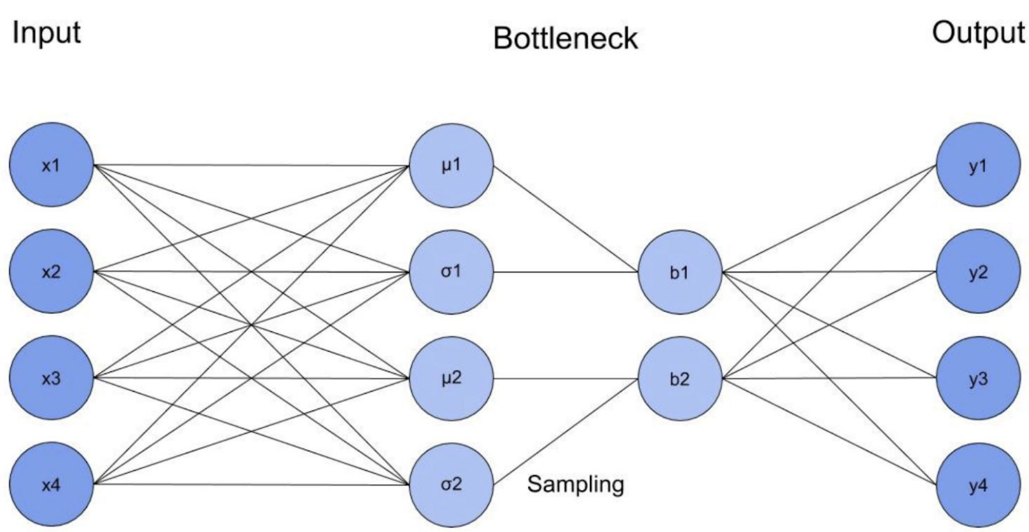

Figure 5: A classical illustration for representing the architecture of a VAE. Figure extracted from (Quentin Bacuet, et al.2019).

This model performs non-linear transformations of the data and its query time is really fast as well once the model has been trained. However, it is more complex to be implemented and the interpretability of the results cannot be provided as easily as in other approaches.

3.3 Recommender System API

Along this section, a complete review of the first release of the Recommender system API will be presented. In the previous section, the set of scenarios or use cases where the system will be launched were commented. In particular: job seeking, digital content (for educational or social purposes) and a social mentoring service were the most concrete cases where the system will attempt to support migrants in their decisions. However, this collection of use cases may undergo modifications in posterior release when new cases or variations of the current ones would be needed.

More specifically, the API is divided into different sections in order to distinguish the kind of operations that are needed regarding the elements involved in the process. In particular, the API is organised as follows:

- Users
- Jobs
- Digital content
- Events
- Places

By considering this approach, the recommender system can be used from both perspectives: a standalone application with its own backend and database that can be hosted independently of the final application where it is planned to be used and as a modular service that can be connected to a more general API which has all the information needed for the analysis.

It is important to remark that the API is in continuous development and integration according to the final user needs and therefore, the next endpoints may undergo modifications.

3.3.1 Users

In Figure 6 , the operations that can be performed regarding the User entity are drawn throughout the Swagger of the component. As expected, typical operations such as create, update or delete users are available to be used during the batch or offline processing of the application.

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

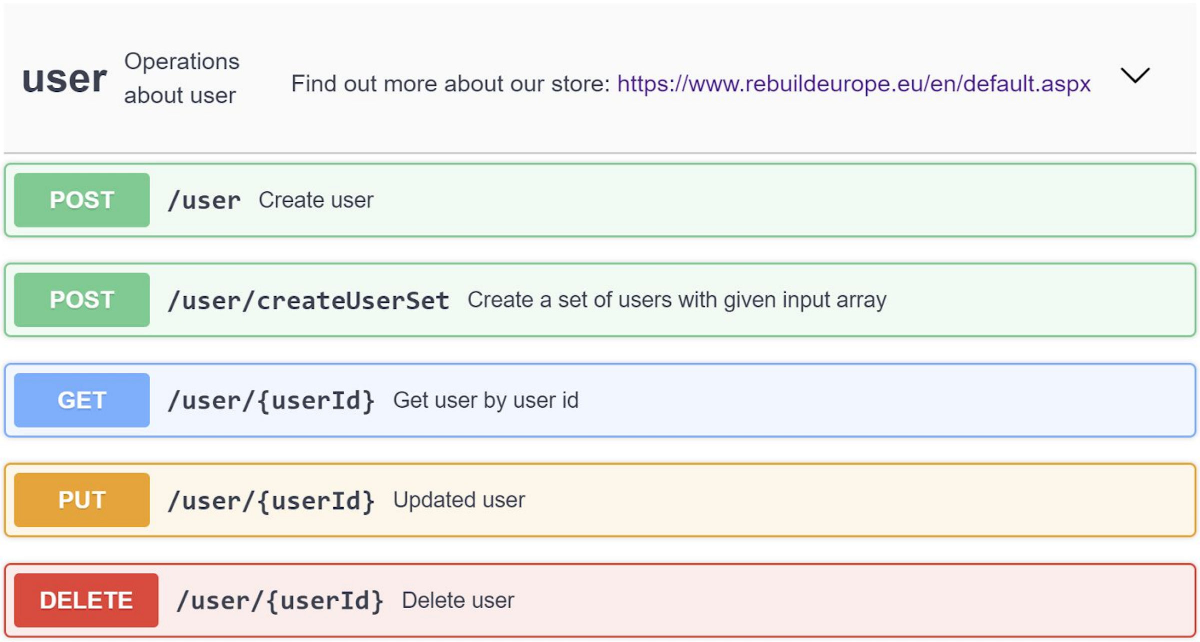

Figure 6: Swagger documentation of the recommender system API regarding Users.

3.3.2 Jobs

In Figure 7, the operations that can be performed regarding the Job entity are drawn throughout the Swagger of the component. As expected, typical operations such as create, update or delete jobs are available to be used during the batch or offline processing of the application. Moreover, a similarity request was included to review similar jobs, this request is crucial for the recommendation part of the application.

In particular, this part of the API will be employed in the job seeking scenario where final users are analysed regarding their competences and skills in order to receive the most adequate jobs for them.

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

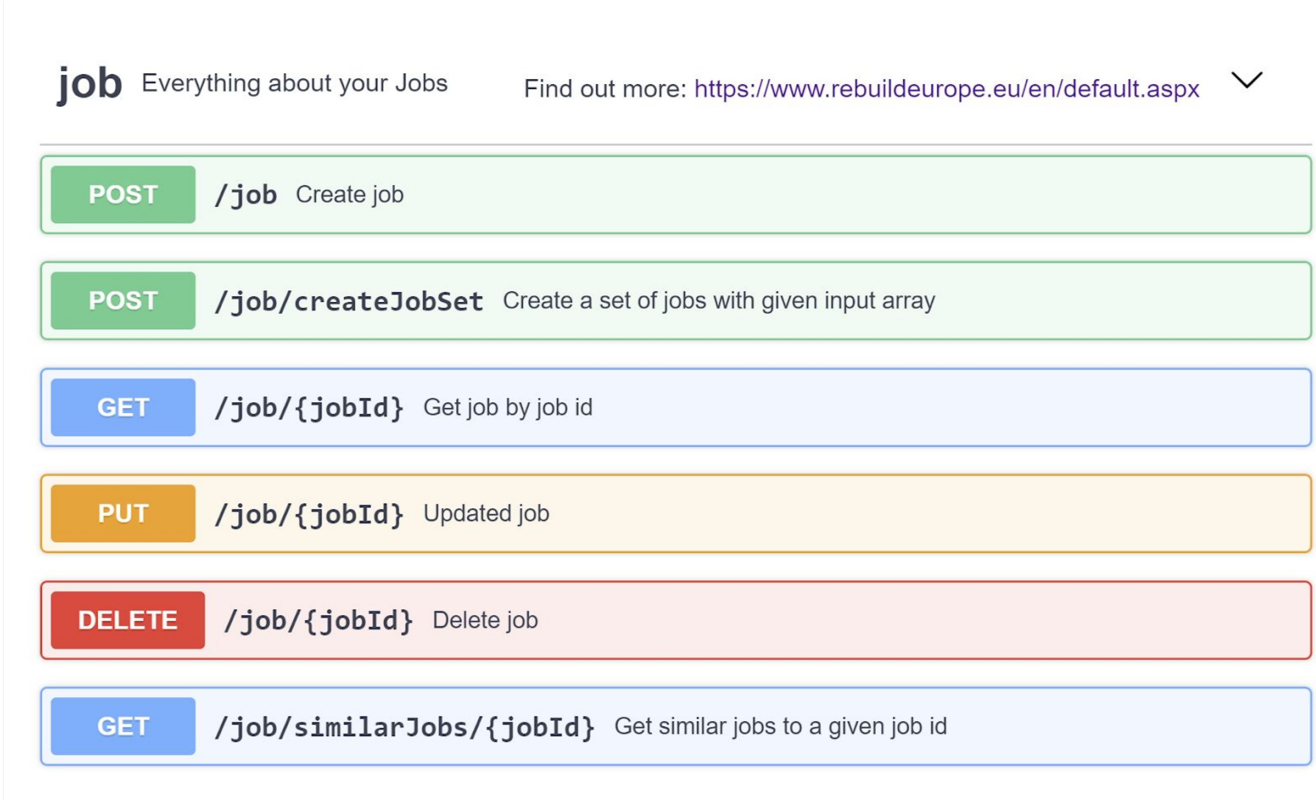

Figure 7: Swagger documentation of the recommender system API regarding Jobs.

3.3.3 Digital Content

In Figure 8, the operations that can be performed regarding the Digital Content entity are drawn throughout the Swagger of the component. As expected, typical operations such as create, update or delete digital content are available to be used during the batch or offline processing of the application. Moreover, a similarity request was included to review similar contents, as in the job scenario, this request is crucial for the recommendation part of the application.

Additionally, a digital content element may be provided for educational purposes, social knowledge about the customs or traditions regarding the arrival country. Therefore, local service providers or any other third party user may incorporate digital content including courses, useful links for interesting local services, video guidelines etc. Anything that may support migrants can be included within this category of data.

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

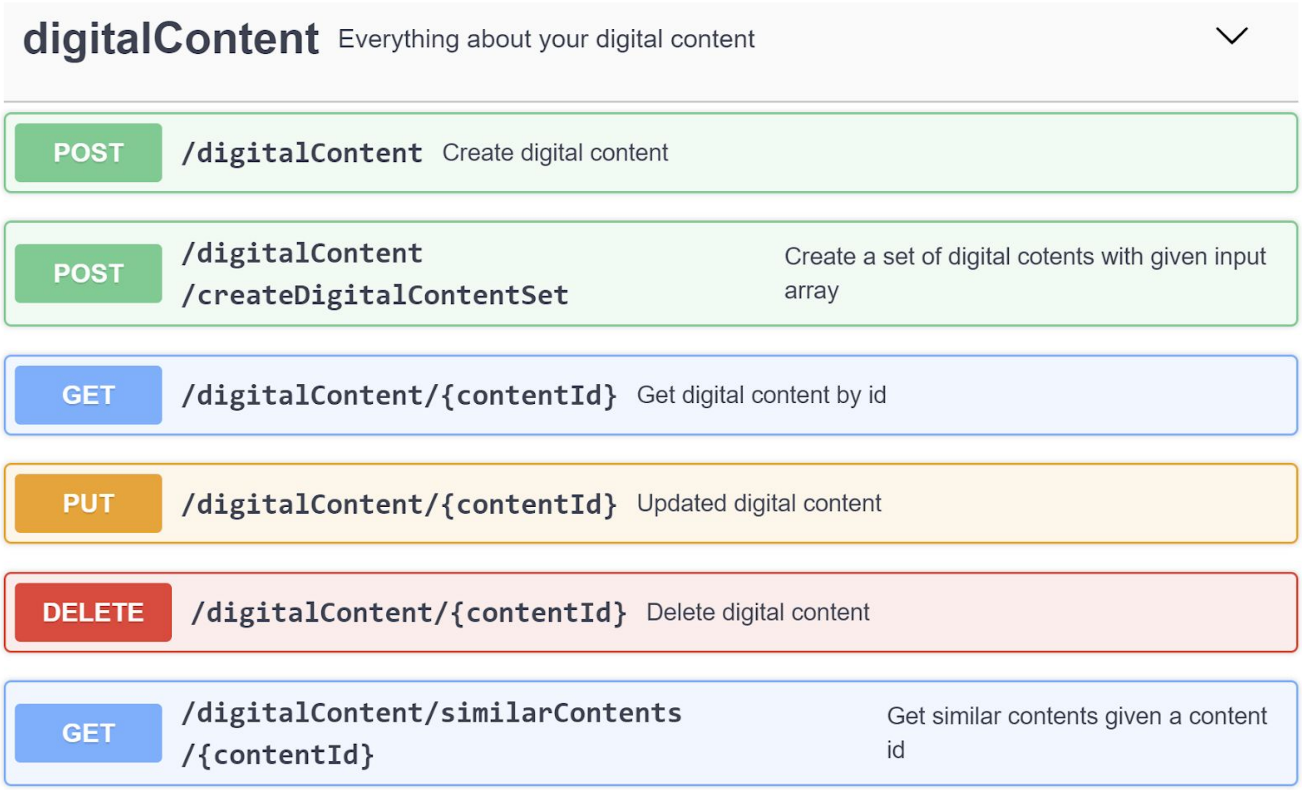

Figure 8: Swagger documentation of the recommender system API regarding Digital Content.

3.3.4 Events

The events that will be available for users can be separated into two categories:

- Those which are retrieved from external APIs such as EventFul[3]
- Those which are created, updated and deleted by local service providers or any other third party user involved within the REBUILD framework.

Moreover, all the available requests regarding the Event object are drawn in the main Swagger of the recommender system as it is illustrated in Figure 9.

[3] https://losangeles.eventful.com/events

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

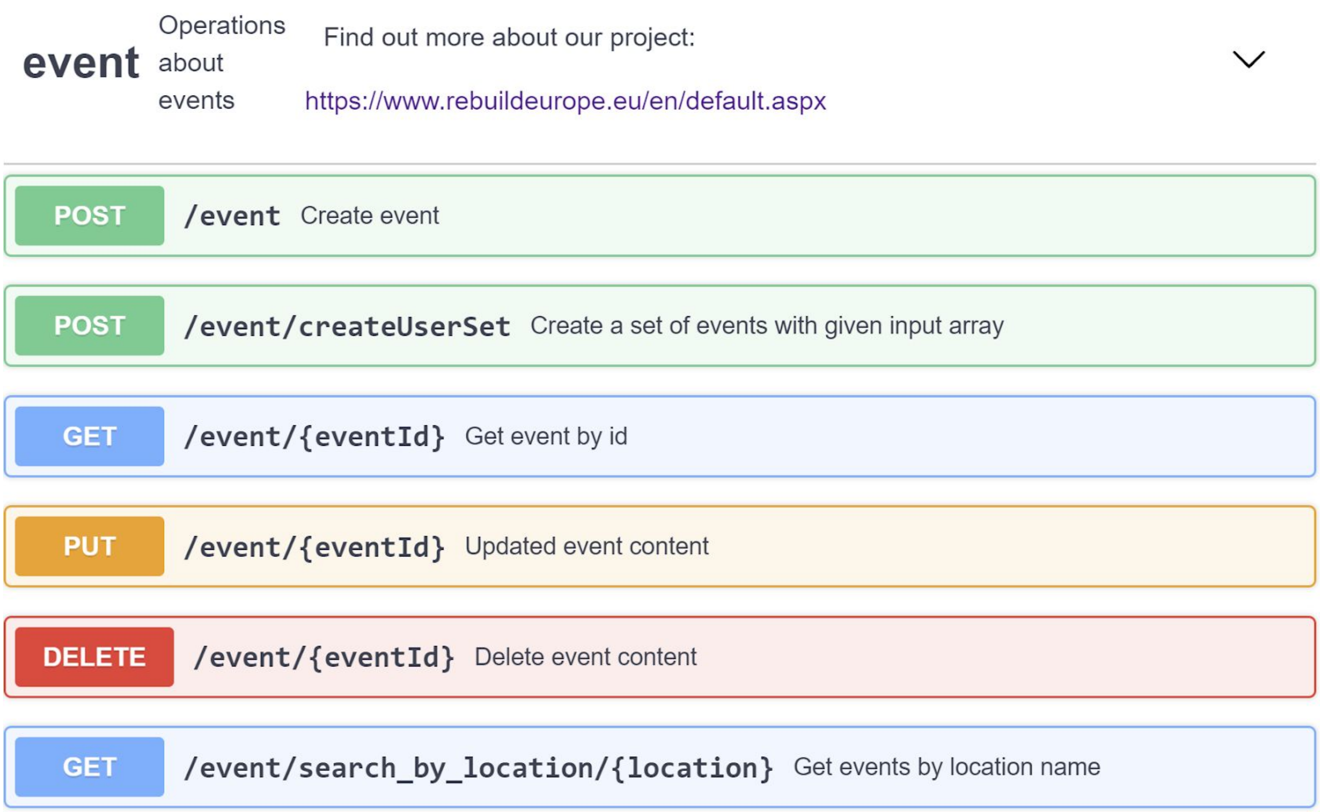

Figure 9: Swagger documentation of the recommender system API regarding Events.

3.3.5 Places

This section describes the part of the API related to interesting places. As in the event case, there are two ways of addressing this situation:

- Add manually places throughout the API, which can be useful for specific local providers or any other third party with the aim to incorporate this knowledge to the system
- Consult external APIs in order to retrieve places of interests for users according to their needs and location.

In Figure 10, all the related requests regarding places of interest are shown. The main goal of this set of requests consists in providing migrants with useful places where they can find support or assistance with some of the main concerns and issues that they will face once they arrive in a new country.

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

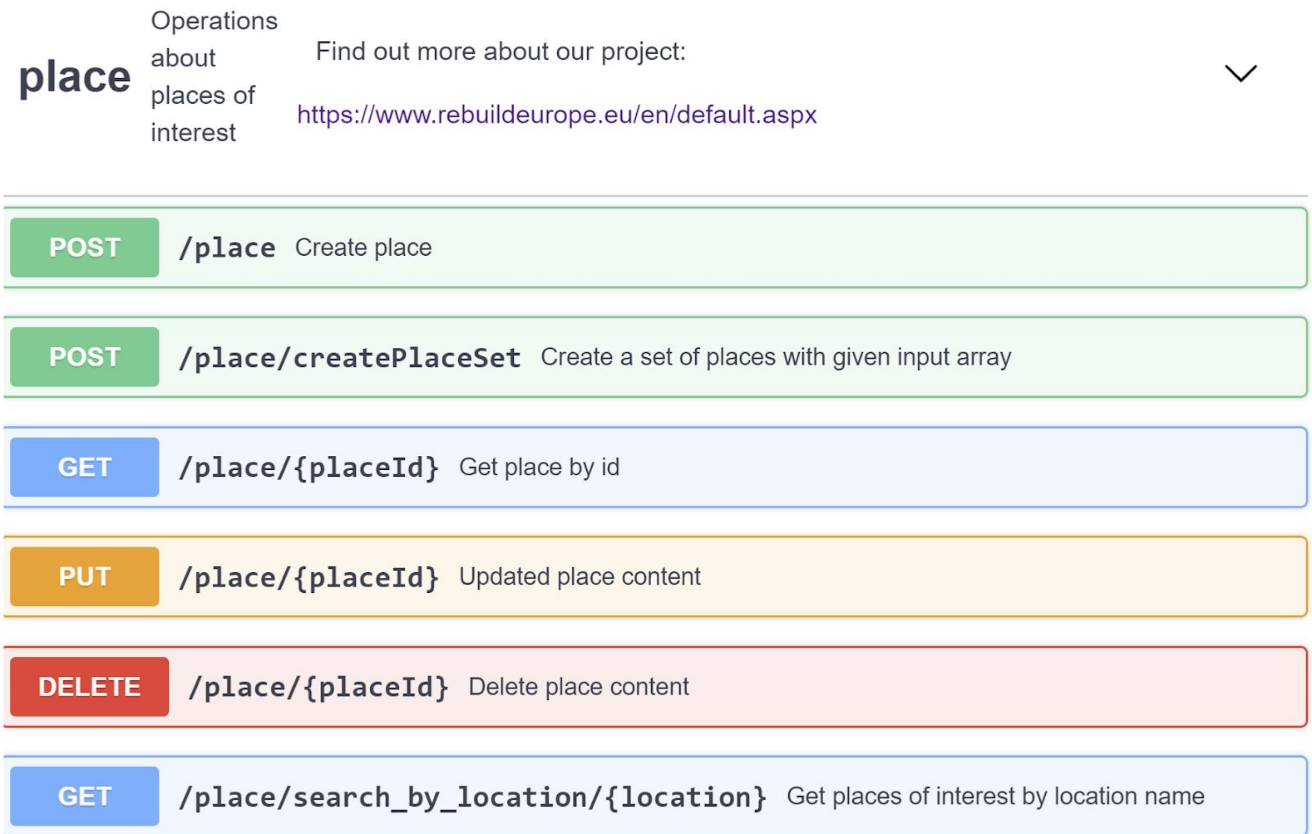

Figure 10: Swagger documentation of the recommender system API regarding Places.

3.3.6 Hybrid Filtering

In this section, all the requests related to the recommendation system execution are shown. In the previous sections we have commented the different elements that can be modified in the component as well as how they are related. However, the API also needs to have some specific requests to perform the different operations involved in the process of generating recommendations.

Moreover, the API has a specific endpoint to call the User Profiling component which will return a set of embeddings representing the profiles of users as well as the clusters according to their similarities. Then, there is an endpoint to run the filtering algorithm which basically will call the skill-matching component or not (depending on the use case) and will generate the set of recommendations after applying a set of algorithms to relate the items and the available users. Finally, there is an endpoint that will retrieve all the recommendations associated with a target user identifier.

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

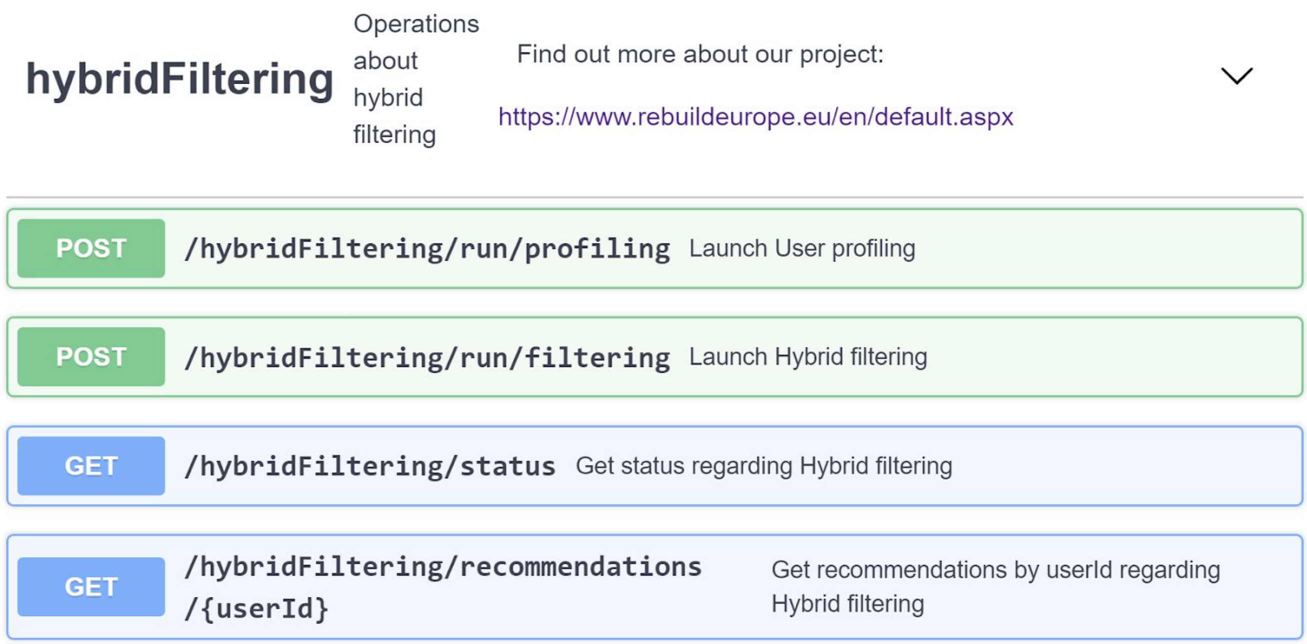

Figure 11: Swagger documentation of the recommender system API regarding the Hybrid Filtering algorithm.

3.4 Use Cases

During this section, we will describe some use cases to illustrate the whole recommender system. To do so, some scenarios will be presented including: job seeking, looking for places of interest and educational content recommendations.

Hence, this section summarizes the main scenarios where the recommender system is planned to be launched and provides an understanding of which kind of recommendations can be provided by the system and how these notifications will be received by the final users of the REBUILD platform.

It is important to recall that since the system is built using a RestFul API, the communication protocol will be both HTTP and HTTPs and all the information from both inputs and outputs are formatted with classical JSON files.

3.4.1 Job Seeking

Job seeking is the main and most interesting scenario where the recommender system will operate. More specifically, the objective of the recommender system in this situation is to assist migrants in the tedious, tiresome process of searching and finding adequate jobs in the new country which can be

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

performed by them according to their skills and competences, not only professionally speaking but personally as well. Figure 12 attempts to illustrate the whole job seeking process.

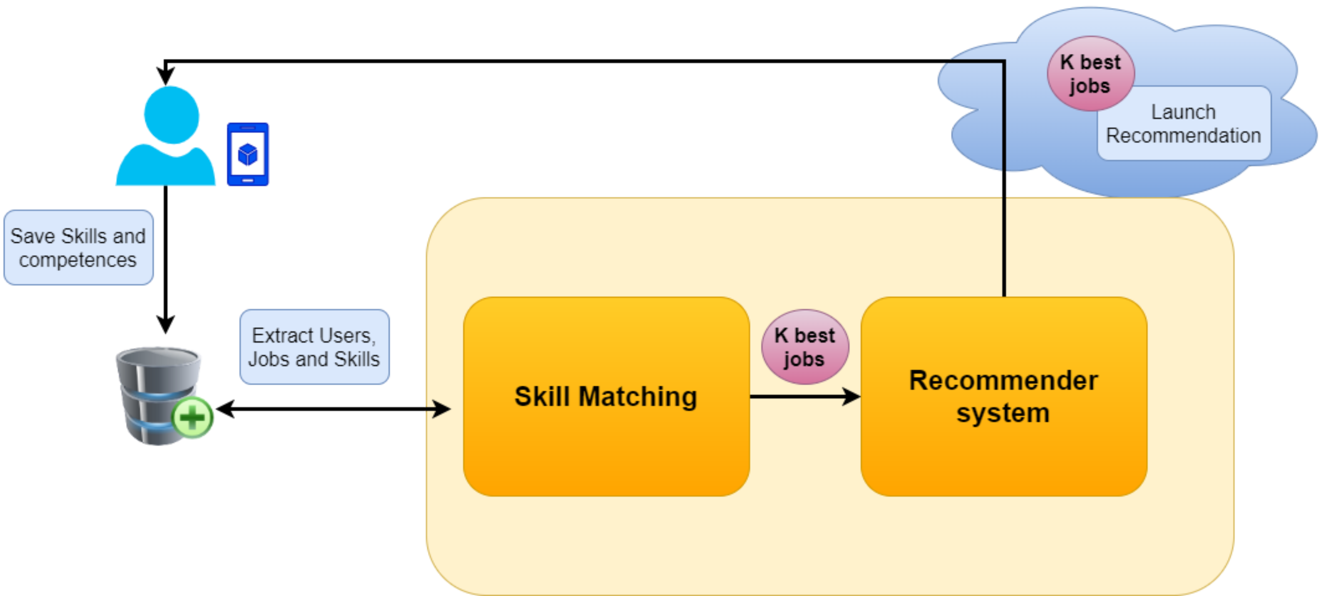

Figure 12: Data workflow regarding the Job Seeking scenario.

3.4.1.1 Inputs

The input in this scenario consists in a set of skills which are rated by migrants according to their level of expertise in each of them. The skills will be added to the system when a new migrant is registered for the first time. At this time, the system asks migrants to include some predefined skills as well as the level of expertise on each of them.

Moreover, the level of expertise is rated in a 1-5 scale, where 1 indicates low expertise whereas 5 indicates the complete opposite indication. An example of this set of inputs is presented in the following table:

skill name	level of expertise
Teamwork	4
Problem-Solving	3
Planning	4
Project management	5

Table 1: Examples of both skills and expertise levels which will be provided by final users.

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

The complete list of Skills that will be available to be selected by users will be continuously increasing in order to incorporate more relevant competencies to the platform with the aim of improving the performance of the recommender system.

3.4.1.2 Outputs

The output of this scenario is a set of K job identifiers indeed which are unique and associated to a single job entity and a matching level which indicates how appropriate the job is to the user. Therefore, to provide the final information to the user, the platform needs to retrieve from the database this set of identifiers and finally provide the final user with the description, the title and all the properties of the Job itself via REBUILD platform. An example of a set of outputs is provided in the next table:

uuid	matching level
7f40c84e-b42f-4867-84d7-e454b110a7be	75%
2df8c63b-29a6-48a2-af94-b6ba378b521c	88%

Table 2: Examples job identifiers (uuid) together with their associated matching level which indicates how a given user fits into such a job.

3.4.2 Places of Interest

Places of interest recommendation refers to providing final users with some advice and places where they can solve some of the pending issues needed to be properly solved in a new country, for instance, where the main hospitals, ambulatories or police stations are in the city etc. Therefore, this is another interesting scenario to support migrants in some basic aspects of life when they first arrive in a new country.
Figure 13 illustrates the general overview of the data workflow that is performed by the system in this scenario. As expected, many of the steps are very similar to previous use cases except for the set of both inputs and outputs which will vary regarding the scenario.

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

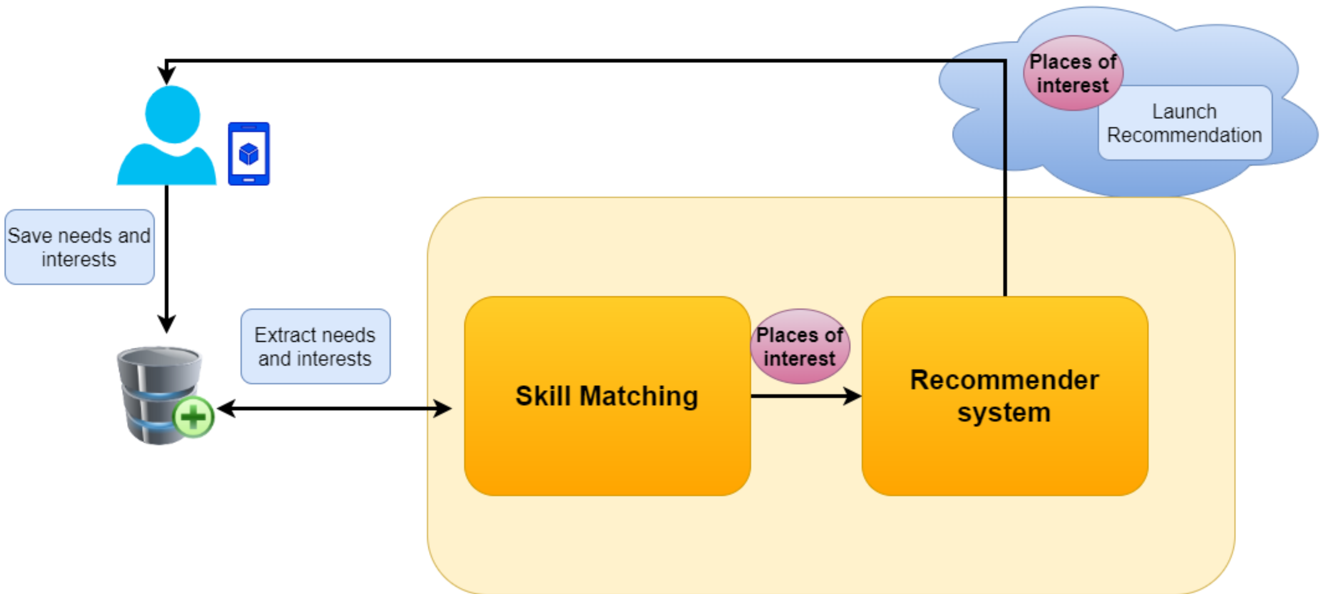

Figure 13: Data workflow regarding the recommendation of places of interest.

3.4.2.1 Inputs

In this case, the inputs of the system are basically two indicators: one more related with the localization of the migrant (in which city is he or she allocated) and the other is related with the necessities that the migrant could have when starting to use the REBUILD application.
In particular, the following table presents a simple example of input that can be interpreted by the system in order to retrieve specific places that may help migrants in some basic procedures.

Needs	**City**	**Country**
Healthcare system	Barcelona	Spain
Personal Identification document	Barcelona	Spain

Table 3: Examples of needs or interests which will be provided by final users.

As it can be observed in the table, a particular migrant has arrived in Barcelona, Spain and he/she needs two main things: having some knowledge about the healthcare system such as where the hospitals are as well as where a personal identification document can be obtained.

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

3.4.2.2 Outputs

The outputs in this case are sent in a very similar way as in the job seeking scenario. However in this case, the outputs are separated by type of place in order to improve the searching of identifiers in the general database. An example of the output is presented in the following table:

uuid	**Type**
7f40c84e-b42f-4867-84d7-e454b110a7be	place
2df8c63b-29a6-48a2-af94-b6ba378b521c	place

Table 4: Example of the output of the system in the place of interest scenario. The output registers the type of place as well as its pertinent unique identifier to make the retrieval of the object in the platform easier.

As expected, the general API of the platform will take the place type property as well as the uuid and it will search in the database for them in order to be retrieved and presented to the final user in a more visual manner.

3.4.3 Digital Content

This third scenario is one of the most relevant ones since it attempts to support migrants with multimedia content and documentation which basically will be used as guidelines or tutorials to manage some specific tasks that a migrant may have to do in his/her first weeks or months in the new country.

The following illustration (Figure 14) shows the data flow of the Digital Content scenario including the interaction between the user and the application as well as the communication among the different components which form the artificial intelligence section of the REBUILD platform.

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

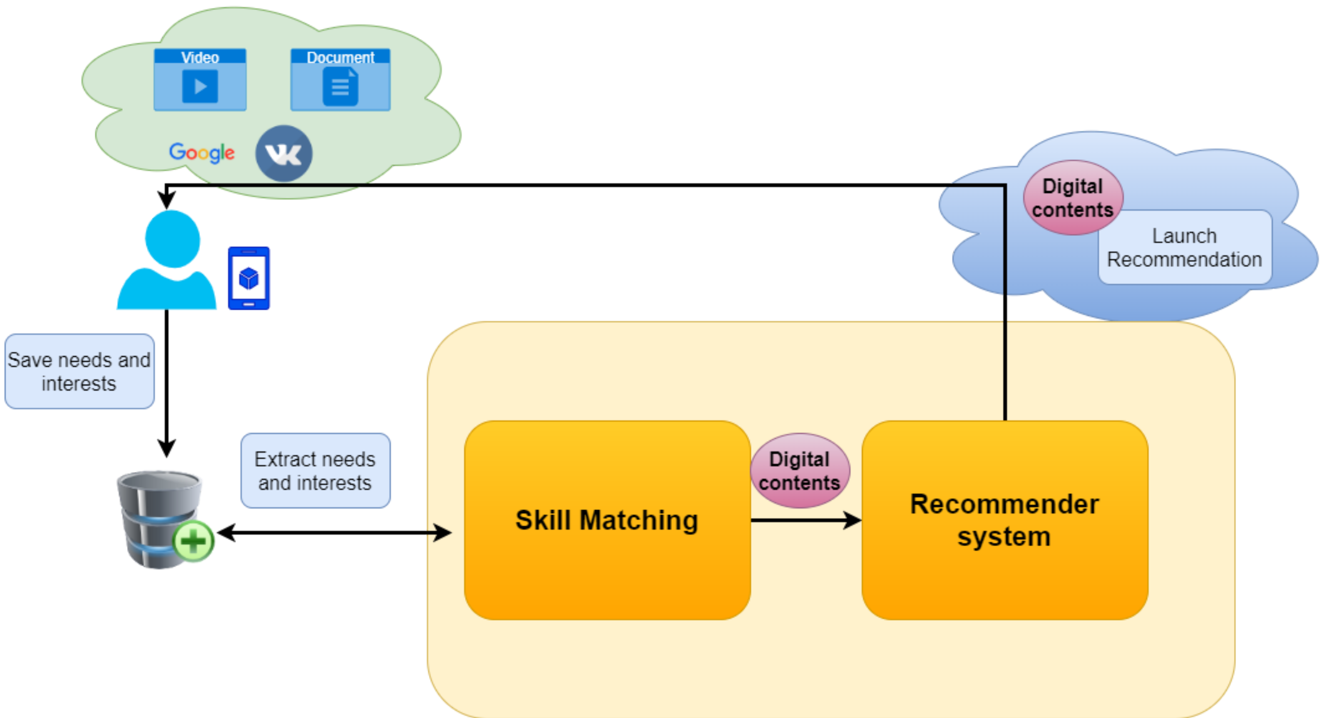

Figure 14: Data workflow regarding the recommendation of digital content such as videos, documents, website URLs and any other kind of information available on the database of REBUILD.

3.4.3.1 Inputs

The input in this scenario is very similar to the Place of interest case, where the user has a certain collection of needs and/or interests about specific domains or topics such as Health, social assistance, food suppliers etc.
The following table aims to present a simple example of which kind of data is provided as input of the recommender system in this particular scenario.

Description	**City**	**Country**	**Type**
Healthcare system	Barcelona	Spain	need
Personal Identification document	Barcelona	Spain	need
Spanish culture	Barcelona	Spain	interest

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

Spanish food	Barcelona	Spain	interest

Table 4: Examples of needs or interests which will be provided by final users.

3.4.3.2 Outputs

The output of the system will consist of a set of identifiers which will correspond to multimedia content or digital content in general. Moreover, this content will be stored in the general database of the system so that the general API of the REBUILD platform will transform this set of identifiers into a more adequate way of visualizing the results to the final user.

An example of a simple output is presented in the following table, where a set of identifiers is returned after calling the analysis methods.

uuid	**type**
2640aa43-e62a-4f95-83f2-ff616e931011	digital_content
e70e1ec5-8c21-41df-a92b-fd5e3b631195	digital_content

Table 5: Examples of identifiers corresponding to digital content information which is available in the REBUILD database.

3.4.4 Events and Social Activities

The last use case that is described in this document deals with the recommendation of events and social activities that may help some migrants to better integrate in the society of the city where they have arrived. Therefore, the goal of this scenario is to encourage migrants to attend to some events that can be provided by either local service providers or any other third party entity.

Moreover, Figure 15 shows an illustrative diagram of the data flow that is followed by the system in order to recommend social events and activities that match with the user profile regarding hobbies, interests, age and many other specifications.

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

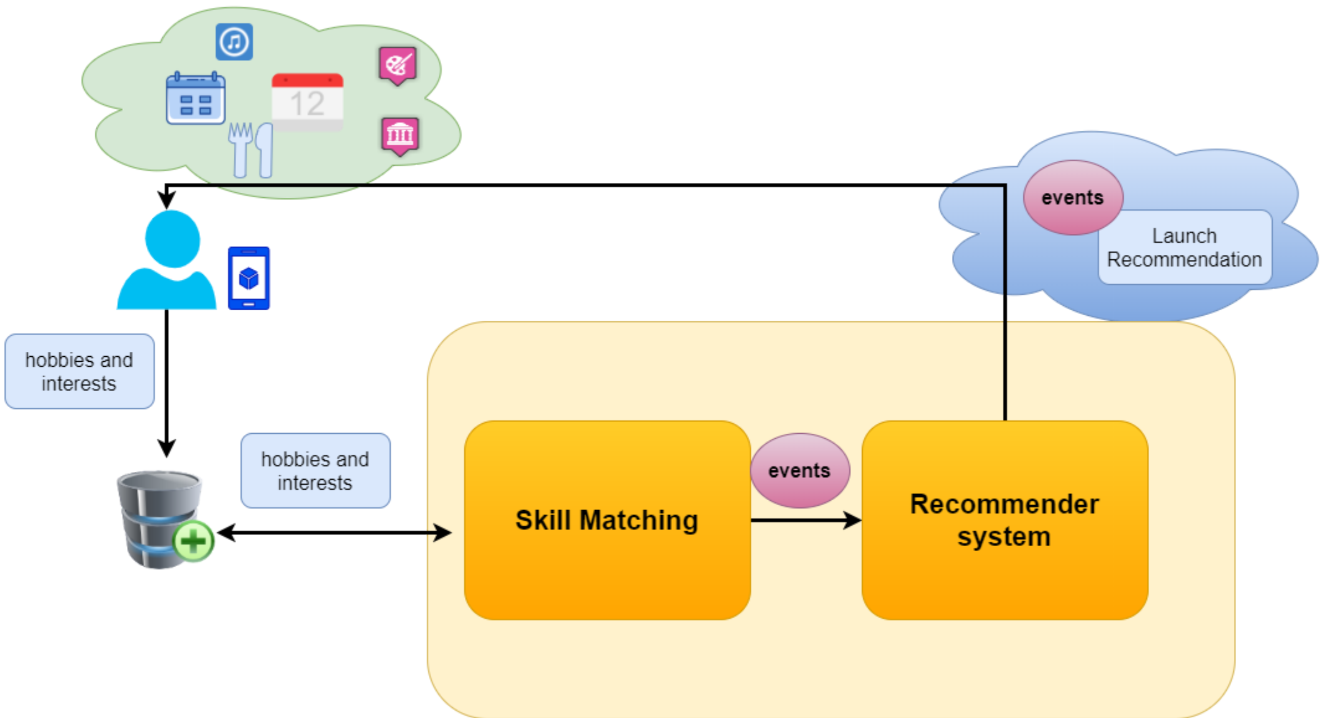

Figure 15: Data workflow regarding the recommendation of social events and activities such as fine-art exhibitions, bars, gigs and music concerts and any other kind of information available on the database of REBUILD.

3.4.4.1 Inputs

In this case, the inputs of the system will consist of a set of hobbies and interests that users have in their personal life such as reading, listening to music, going out etc. The following table summarizes an example of this kind of information.

description	**type**
Spanish language	interest
Science	interest
Playing the guitar	hobby
Reading	hobby

Table 6: Examples of interests and hobbies that users may have. This information will be provided when a particular user is registered in the REBUILD platform for the first time.

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

3.4.4.2 Outputs

The output in this scenario will consist of a set of identifiers which will correspond to events and activities that are available within the REBUILD application. Thus, as in other cases, the recommender system will compute the algorithms and will return a collection of identifiers and the general API of REBUILD will transform these identifiers into the event objects that contain all the details of the activities indeed.

An example of this sort of output is presented in the following table:

uuid	**type**
4ed09c7b-278b-4b63-ba62-0284dddc8b3a	event
3f3917f7-72c4-4eb0-a89a-cb3f92ad0841	event

Table 7: Examples of unique identifiers which correspond to specific events that are registered within the REBUILD platform.

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

4 CONCLUSION

This deliverable describes the work done regarding the task 3.3 from WP3. The aim of this document is to present the techniques that are employed to perform the recommendation of services, digital contents or jobs to migrants who have arrived to a new country and thus, they would need some support in many daily aspects of both professional and personal life.

Moreover, the deliverable details all the work that has been performed in other fields and applications with the purpose of emphasizing the different approaches and solutions that can be assessed within the REBUILD framework including collaborative, content-based or hybrid filtering.

Furthermore, after reviewing and researching the different techniques, the hybrid filtering was selected to be the optimal solution in this project due to the possibility of combining not only the information from users (user profile) but also the metadata corresponding to the different items that the database of the project has including digital content, jobs, events or places.

The document describes the initial API provided by the recommender system which can be used as a standalone application or as a component of a more complete platform as in our case. The description of the API incorporates the main endpoints that are proposed at first. However, since the project has a user-driven nature, some of the requests may undergo slight modifications during the circle of the life of the project.

Finally, this deliverable presents a collection of use cases and scenarios, where the inputs and the outputs that are proposed at first are depicted together with some illustrations to improve the understanding of the data flow in each specific case.

As it was already mentioned, this deliverable presents the first architecture and development of the recommender system which can be subjected to minor changes according to both the feedback of users impacting any component within the platform of REBUILD.

The future work will be mainly devoted to adjust the type of recommendation to the user's needs. For this reason, all stakeholders (IT team, LSP and users) will collaborate for the appropriate setting of the recommendations. Also, the interaction with the Skill matching module and User Profiling for the pipeline integration will be carried out.

5 REFERENCES

Osadchiy, T., Poliakov, I., Olivier, P., Rowland, M., & Foster, E. (2019). Recommender system based on pairwise association rules. Expert Systems with Applications, 115, 535-542.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The adaptive web (pp. 325-341). Springer, Berlin, Heidelberg.

Shaw, G., Xu, Y., & Geva, S. (2010, June). Using association rules to solve the cold-start problem in recommender systems. In Pacific-Asia conference on knowledge discovery and data mining (pp. 340-347). Springer, Berlin, Heidelberg.

Pajuelo-Holguera, F., Gómez-Pulido, J. A., Ortega, F., & Granado-Criado, J. M. (2020). Recommender system implementations for embedded collaborative filtering applications. Microprocessors and Microsystems, 102997.

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002, August). Methods and metrics for cold-start recommendations. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 253-260).

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. Expert Systems with Applications, 41(4), 2065-2073.

Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., & García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain. Expert Systems with applications, 39(12), 10990-11000.

García-Sánchez, F., Colomo-Palacios, R., & Valencia-García, R. (2020). A social-semantic recommender system for advertisements. Information Processing & Management, 57(2), 102153.

Kiran, R., Kumar, P., & Bhasker, B. (2020). DNNRec: A novel deep learning based hybrid recommender system. Expert Systems with Applications, 144, 113054.

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 52(1), 1-38.

Lukasiewicz, T., Miao, Y., & Xu, Z. (2017). Tag– Aware Personalized Recommendation Using a Hybrid Deep Model.

Zhang, S., Yao, L., & Xu, X. (2017, August). AutoSVD++ An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 957-960).

Yu, W., Zhang, H., He, X., Chen, X., Xiong, L., & Qin, Z. (2018, April). Aesthetic-based clothing recommendation. In Proceedings of the 2018 World Wide Web Conference (pp. 649-658).

Xie, R., Liu, Z., Yan, R., & Sun, M. (2016). Neural emoji recommendation in dialogue systems. arXiv preprint arXiv:1612.04609.

Xie, W., Ouyang, Y., Ouyang, J., Rong, W., & Xiong, Z. (2016, December). User occupation aware conditional restricted boltzmann machine based recommendation. In 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 454-461). IEEE.

Zheng, Y., Tang, B., Ding, W., & Zhou, H. (2016). A neural autoregressive approach to collaborative filtering. arXiv preprint arXiv:1605.09477.

Wang, X., Yu, L., Ren, K., Tao, G., Zhang, W., Yu, Y., & Wang, J. (2017, August). Dynamic attention deep model for article recommendation by learning human editors' demonstration. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 2051-2059).

He, X., He, Z., Du, X., & Chua, T. S. (2018, June). Adversarial personalized ranking for recommendation. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 355-364).

Choi, S., Ha, H., Hwang, U., Kim, C., Ha, J. W., & Yoon, S. (2018). Reinforcement learning based recommender system using biclustering technique. arXiv preprint arXiv:1801.05532.

Gao, J., Pantel, P., Gamon, M., He, X., & Deng, L. (2019). Modeling interestingness with deep neural networks.

Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016, August). Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 353-362).

Shuyu Luo et al. (2018, December) Introduction to Recommender System: Approaches of Collaborative Filtering: Nearest Neighborhood and Matrix Factorization. In Medium. Information was retrieved from https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26.

Amato, F., Moscato, V., Picariello, A., & Piccialli, F. (2019). SOS: A multimedia recommender System for Online Social networks. Future generation computer systems, 93, 914-923.

Xianxian H. (2019, May). Knowledge Graph (KG) for Recommendation Systems:An introduction of how to integrate Knowledge Graph with Recommendation System. In Medium. Information was retrieved from https://medium.com/@hxianxian/knowledge-graph-kg-for-recommendation-system-8fe2c6cd354.

Wang, H., Zhang, F., Xie, X., & Guo, M. (2018, April). DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 world wide web conference (pp. 1835-1844).

Quentin Bacuet (2019, April) How Variational autoencoders make classical recommender systems obsolete. In Medium. Information was retrieved from https://medium.com/snipfeed/how-variational-autoencoders-make-classical-recommender-systems-obsolete-4df8bae51546

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: REBUILD recommendation environment and follow-up

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