This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 870811

SPICE

Social cohesion, Participation, and Inclusion through Cultural Engagement

D3.6 Prototype community recommender

Deliverable information	
WP	WP3
Document dissemination level	PU Public
Deliverable type	DEM Demonstrator, pilot, prototype
Lead beneficiary	UH
Contributors	UCM
Date	28/04/2022
Document status	Final
Document version	1.0

INTENTIONALLY BLANK PAGE

Project information

Project start date: 1st of May 2020

Project Duration: 36 months

Project website: https://spice-h2020.eu

Project contacts

Project Coordinator

Silvio Peroni

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

Department of Classical Philology and Italian Studies – FICLIT

E-mail: silvio.peroni@unibo.it

Project Scientific coordinator

Aldo Gangemi

Institute for Cognitive Sciences and Technologies of the Italian National Research Council

E-mail: aldo.gangemi@cnr.it

Project Manager

Adriana Dascultu

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

Executive Support Services

E-mail: adriana.dascultu@unibo.it

SPICE consortium

No.	Short name	Institution name	Country
1	UNIBO	ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA	Italy
2	AALTO	AALTO KORKEAKOULUSAATIO SR	Finland
3	DMH	DESIGNMUSEON SAATIO - STIFTELSEN FOR DESIGNMUSEET SR	Finland
4	AAU	AALBORG UNIVERSITET	Denmark
5	OU	THE OPEN UNIVERSITY	United Kingdom
6	IMMA	IRISH MUSEUM OF MODERN ART COMPANY	Ireland
7	GVAM	GVAM GUIAS INTERACTIVAS SL	Spain
8	PG	PADAONE GAMES SL	Spain
9	UCM	UNIVERSIDAD COMPLUTENSE DE MADRID	Spain
10	UNITO	UNIVERSITA DEGLI STUDI DI TORINO	Italy
11	FTM	FONDAZIONE TORINO MUSEI	Italy
12	CELI	MAIZE SRL	Italy
13	UH	UNIVERSITY OF HAIFA	Israel
14	CNR	CONSIGLIO NAZIONALE DELLE RICERCHE	Italy

Executive summary

D3.6 provides a report on the development of a prototype social recommender in SPICE. To date, after studying the user modelling requirements of the different case studies, a user model and community model data structures and reasoners were defined and implemented, as well as APIs for accessing the data in the linked data hub [LDH]. The user and community model data is used by the social recommender for considering what information to present to the visitor. The social recommender is triggered by the case study visitors' guide/system to suggest a content to the current visitor according to the scenario guidelines - given the user model, search for similar/different content provided by similar/different users in similar/different communities. The main challenge we faced in this stage is the need to adapt to the evolving requirements of the case studies, as they continuously gain knowledge during the design of the case studies. This adds uncertainty about specific requirements of case studies from the recommender that will have to be addressed in the 3rd year. Moreover, the unstable situations in the museum case studies due to the COVID-19 issues, could eventually lead to some readjustments of the original case studies. An initial generic recommender infrastructure was designed and implemented. The details of the technology are described in this deliverable. In the next stage the specific internal reasoning mechanisms of the prototype social recommender will be further developed and adapted according to emerging requirements of the case studies.

Document History

Version	Release date	Summary of changes	Author(s) -Institution
V0.1	20/03/2022	First draft	Tsvi Kuflik and Alan Wecker (UH), Guillermo Diaz (UCM).
V0.2	20.4.2022	2nd draft	Tsvi Kuflik and Alan Wecker (UH).
V0.3	23/4/2022	3rd draft	Alan Wecker (UH), Guillermo Diaz (UCM).
V1.0	28/04/2022	Final version submitted to REA	UNIBO

Table of Contents

1. Introduction

WP3 Deliverable 3.6, the social recommender, focuses on using the technological infrastructure that is developed in WP3 (Deliverables 3.1 and 3.2) and providing content recommendation to individual visitors with the abstract SPICE aim - to promote social cohesion - by allowing the users to explore and experience content that is similar or different to their point of view and produced by members of similar or different communities - according to the systems curator guidelines. Deliverables 3.1 and 3.2 enable reasoning about individuals, their characteristics and preferences, the explicit communities they identify themselves with and the implicit communities that evolve from analysing content contributed by these individuals. Deliverable 3.6 WP3, uses the user and community models and scripts (guidelines/instructions for activities, generated by WP6) for guiding the process of content recommendation to users. When activated by the museum system, the recommender searches the LDH for content to be presented to the user according to predefined guidelines.

D3.6 describes the design and development of the prototype social recommender. The document is organized as follows:

It starts with an abstract description of the social recommender and its services. Following is a description of the internal and external links of WP3 and the role of the social recommender there. Finally, there are conclusions reached so far. An appendix includes detailed APIs, usage examples and code of the different services.

2. Social Recommender (SR)

2.1 Motivation and justification

Normally, recommender systems in general and recommender systems for cultural heritage in particular, try to maximize the satisfaction of each individual choice by the user (Ricci et al. 2015). Alternatively, this can be motivated by the goals of, not necessarily that of the end user, but rather by the goals of organizations trying to promote a certain item. In addition, in recent years a broader view of recommendation goals has been established such as diversity, and serendipity (Kotkov et al., 2020). In SPICE, we aim to apply novel recommendation techniques to advance additional goals of promoting social cohesion and inclusion. Our approach includes the community feeling (Adler, 1964) as a useful component of the recommender algorithm that considers not only the individual preferences but also the individuals' awareness of being part of a certain community and how this feeling affects individuals' preferences when dealing with a set of communities in the social world. Community feeling is beneficial both for the individual and for society. Our approach also includes use of techniques of Citizen Curation (Ellwood et al., 2018; O'Neill, 2017; Robinson & Carletti, 2019). In the SPICE project we define Citizen Curation (Bruni et al., 2020) as citizens applying

curatorial methods to archival materials available in heritage and memory institutions as well as to items depicted in exhibitions in order to develop their own interpretations, share their own perspective and appreciate the perspectives of others. While our approach will clearly be to the benefit of society, we believe that this also reflects the values of the individual user who wishes to better understand their context and environment. Our goal here is that each member of society feels that he has a part of the society's cultural heritage in some manner. To clarify this means while we don't come to a common understanding or interpretation, but rather each individual will understand and perhaps even empathize towards other members of his society's divergent views. Hopefully by citizen curation i.e. allowing people to give their own interpretations we can increase inclusion; and by showing divergent views in an empathetic manner, we can increase social cohesion. To illustrate our ultimate goal, we describe the following scenario:

Lara decides to attend a Citizen Curation activity at a museum she is visiting. The activity involves selecting an artwork from the museum's collection, and providing her own interpretation. Afterwards, after the system considers her interpretation, Lara is notified of an interpretation of the artwork contributed by someone from a different social group (this is accomplished via our social recommender). The interpretation is a personal story prompted by the artwork recorded as audio. The story is accompanied by comments responding positively to the story contributed by people in the first social group. Lara decides to listen to it. Before the audio recording starts, Lara is encouraged to imagine how the storyteller feels about what happened. The story is very different from her interpretation of the artwork. She adds her own comment after listening.

In the SPICE project, user models represent the individuals that are interacting with the system. Community models represent users' community, whether explicit or implicit (explicit – explicitly reported by the user, implicit – created automatically by the community modeler according to the users' generated content). These models are key elements (together with the curator's guidelines - scripts) used to guide the process of content recommendations to individuals, taking into consideration individual and community interests, as well as script guidelines (developed by WP6), to search and identify relevant users' contributions, to provide alternative interpretations of objects, and, as described above, to promote the social contagion among users and to emphasize the similarities and differences within and across communities.

In this document we review the initial studies carried out to identify how a social recommender may be used by the different case studies. The information gathered through a series of project meetings guided the development of the interim community model and a communication protocol to activate its services.

With respect to the user model, deliverable D3.6 describes the interim community recommender for SPICE. It is the result of ongoing interactions with case studies leaders and other work package leaders.

2.2 Description

The purpose of this component is to provide social recommendations of user generated content to aid in the implementation of the interpretation-reflection loop of WP2; primarily reflection (Deliverable D2.4). You can choose similar viewpoints by different communities to engender inclusion and use different viewpoints by similar communities to try and engender cohesion. The recommendations are based on similar and dissimilar views of topics and subjects and material from both similar and dissimilar communities. Views of the subject are determined by the Semantic Analyzer. This is stored in the UM for use by the CM who generate similar and dissimilar communities which is then used by the social recommender to provide recommendations.

The API consists of a single call which attempts to do as much as possible for the developer in providing recommendations of user generated content. Communities can either be explicit or implicit

The idea is to give the script designer the possibility to find people who have common background but have either a different or same opinion or alternatively people who think alike but have a different or same opinion

A typical scenario is that a recommendation of user generated content (opinion, curation and so on) is requested. User generated content is chosen based on the analysis of the Semantic Analyser (similar or dissimilar) it is then filtered by content belonging to users of certain communities.

3. Interaction of the Social Recommender within Work Package 3

Within WP3, the role of the social recommender is to provide recommendations for content to the visitor whenever there is a request. It relies on the analysis done by the user and community modellers – the user and the communities the user belongs to and on the guidelines regarding the type of content to provide. Reasoning on the guidelines and the models, the recommender selects the relevant content (or a set of candidates) and present them to the visitor, as illustrated by figures 1 and 2.

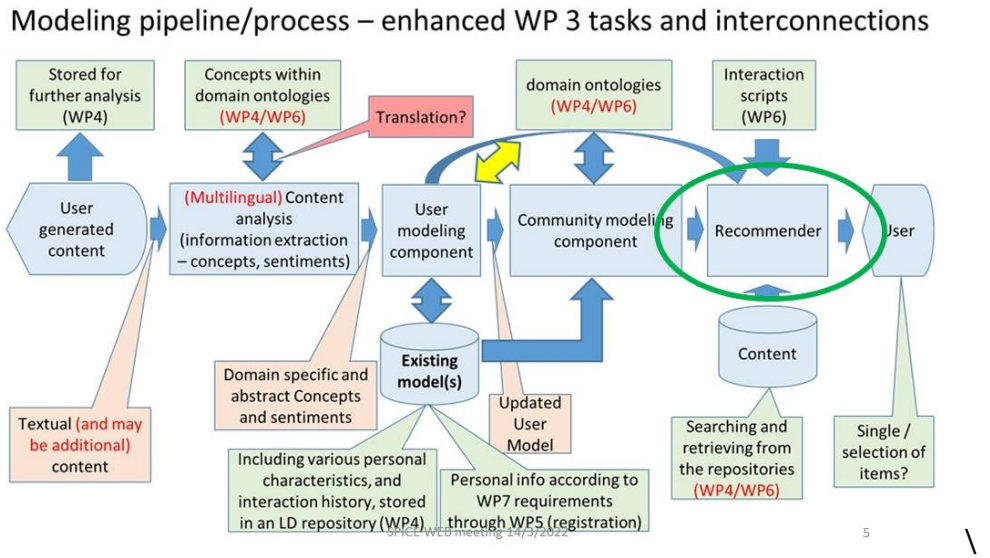


Figure 1: The interaction of the social recommender with WP3 components and external components: user modelling and community modelling data is received from the modelling components, guidelines are defined externally and the content is selected according to the reasoning of the recommender and suggested to the visitor

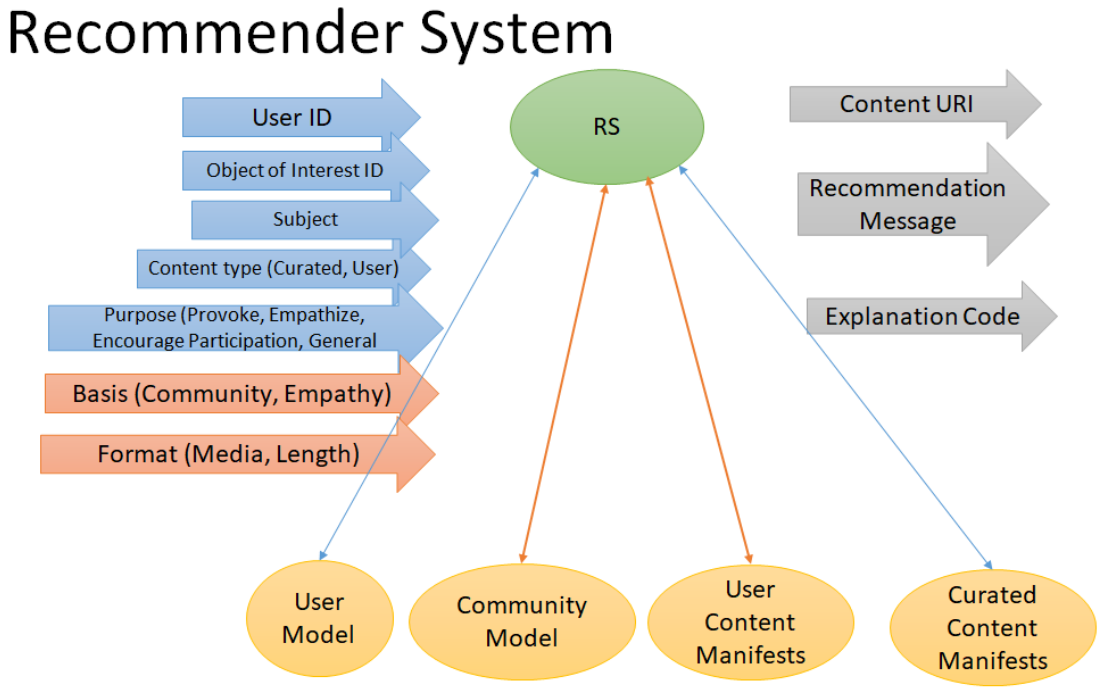


Figure 2: The social recommender process illustration: user and community information is used for searching and delivering content to the user, the user's feedback is sent back to the user and community model – for updating them.

4. Interaction (of the Social Recommender) with other Work Package

With respect to external interaction, the social recommender has indirect interaction with external WPs (most of these interactions done by other parts of WP3). For content recommendation it has to search the LDH for an appropriate content, once a recommendation is defined (e.g. the characteristics of the content to be suggested to the visitor). Here is diagram of the interactions of WP3 with the other packages, where at the very bottom there is the sequence diagram representing a recommendation process:

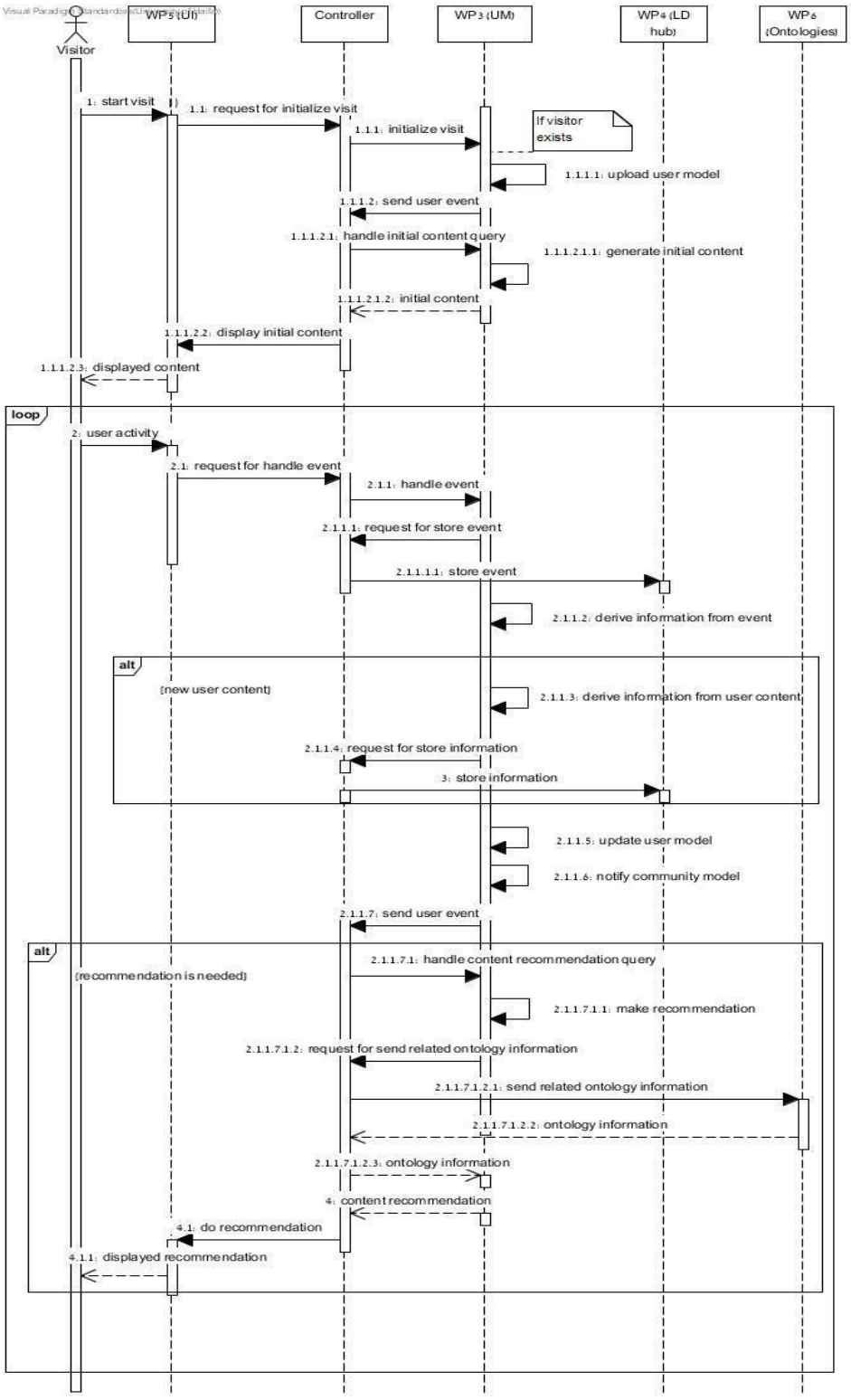


Figure 3: The interaction of the social recommender with external WPs

With respect to the needs of the individual case studies (WP7), meetings were held with each of the case studies (IMMA, GAM, Madrid, Hecht, DMH) to understand their needs and use of the user and community models and implications for the planned recommender. This is summarized in table 1.

Table 1: SPICE case studies their community and recommendation subjects

Case Study	Explicit Community Characteristics	Implicit Community Characteristics	Subjects for recommendation
IMMA	Specific invited Communities (LGBT+, Blacks, etc...	Based on Artwork Interpretation	Artworks
GAM	Schools, Deaf Teenagers	Level of Physical Challenges, similar responses	Artworks
Madrid	Schools	Attitudes toward climate change, Age, Rural vs Urban	Climate Change, Bio-Diversity, Sustainability
DMH	Age, Address	City vs Non City Dwellers (Need to see scenario of asylum seekers)	
Hecht	Schools	Religion, Nationality, Religiosity, Values	Galillee Camapaign, Josephus, Museum Curation

5. Demo

Interactions between UM and CM and the CM and RS (Recommender System). Using one of the case study scenarios (specifically Hecht) as a basis for Demo

5.1 Definitions

The Community Model distinguishes between two types of communities: ***explicit communities*** that are defined on demand by museum curators, and ***implicit communities*** that are discovered based on user **personal attributes** and the information extracted from user interactions (**interaction attributes**).

Here is an example of communities for the Hecht Case Study

1. Explicit communities - communities decided by the case study either by participation or questionnaire or info filled in by a mediator
    a. E.g., for Hecht- School, Religiosity, Political stance
    b. This information will be contained in the UM demographics category or the UM user object)

2. Implicit communities - communities determined by interaction of the user (usually based on a derived value of users' sentiments/emotions about a certain subject (entity) or even an explicit direct asking of a user
    a. E.g., for Hecht sentiment about Josephus Flavius, The Galilee Campaign as part of the First Roman-Jewish War, Museum Curation neutrality (as connected to Galilee Campaign exhibit in the Hecht museum) also AOT (Actively Openminded Thinking) RHMS (Historical Relevancy) AOT Change, RHMS Change
    b. This can be found in the interests, traits, beliefs categories of the UM

5.2 Phase 1) UM to CM

a. After collecting information in a phase it will call the CM that info for a certain user is ready

POST /users/user1/update-generated-content (body content contains an array of objects with {user1,propertyname,propertyvalue...})

As part of collecting information the app the UM calls itself when UGC (User Generated Content (UGC) is created this information can be enhanced by **tags** (which is user's self-description of the UGC) or **metadata** (which is sentiment or emotion detected in the UGC by the Semantic Annotator) which can be enhanced when information is missing to info from a parent UGC. The tags and metadata are used to update UM (which is part of the info the CM uses) . In the future maybe add **ratings** by other users. The UGC Info will contain an "origin" id in the schema which will point to original object

5.3 Phase 2) CM to UM, RS to CM, UM, UGC

When a recommendation is requested RS calls UGC LD-HUB to find content that is similar/dissimilar to user's opinion (The SPARQL query to do this is configured on a case by case basis)

RM calls CM to find what are similar/dissimilar communities to those of the user

/users/{user-id}/communities

Communities that a user belongs

1. Similarity API

RM calls CM to find users of those communities requested (using the k –parameter)

/communities/{community-id}/similarity or

/communities/{community-id}/disimilarity

2. Get a list of users

This is in the communities' object returned by the previous call

RM filters answers from UGC to users belonging in desired community

5.4 Future

RM can play with k of similarity and similarity of views to get a reasonable number of suggestions. In the future RM may call CM for explanation why this content was returned and asked to modify his community.

5.5 General Structure

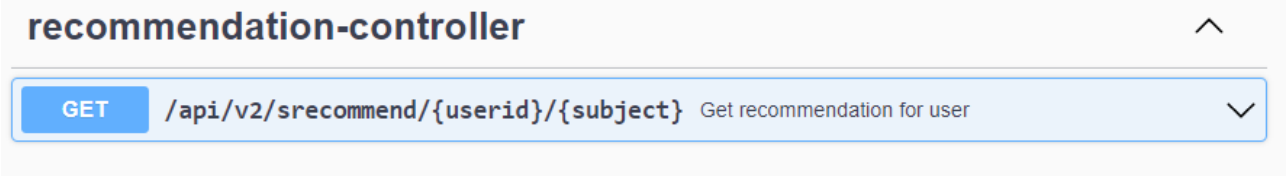
The deliverable consists of prototype code of REST APIs[1], built using the SPRING boot framework[2] and this document which describes the role which the user model plays in the overall project. In terms of deployment in the project, the intention is that each case study would instantiate its own version of the REST server. In addition, a REACT frontend[3] was developed which gives an example of how to use the REST services.

[1] https://en.wikipedia.org/wiki/Representational_state_transfer
[2] https://spring.io/projects/spring-boot
[3] https://reactjs.org/

5.6 Accessing the Social Recommender

The following is the REST API:

The recommendation controller is used to request appropriate user generated content



Recommend(Userid, Subject, Configuration) which return a ranked list of User Generated Content ids

Parameters

Userid - for who is the recommendation

Subject- name for User generated content items with this entity tag

Configuration

SubjectSentimentSimilarity - Same as user (SU), Different from User (DU), Don't care (DK)

And (

ExplicitCommunitySimilarity - (community name) SU, DU, DK

or

ImplicitCommunitySimilarity - (community name) SU, DU, DK

)

Note: The purpose is to give the script designer the possibility to find people who have common background like me (same explicit community) but have a different/same opinion on the subject (as provided by Semantic Analyzer) or people who think like me (same implicit community) but have a different/same opinion (as provided by Semantic Analyzer)

5.6.1 Example of Use

The demo is a modified verson of the *studentmgr* app from the Hecht case study (https://hspice.haifa.ac.il/studentmgr2). In the app we allow the teacher to find material that is similar/dissimilar to those of a particular student's viewpoint.

Since React doesn't make REST calls directly we use the axios4 library to wrap the REST API calls. In Appendix 9.3 we show the wrapping of the REST call for the Social Recommender.

6. Conclusions and future work

In general, for task 3.6 the goals set for the first year were achieved. The heterogeneity of the case studies (which is a good thing) posed a challenge on the development of a generic social recommender required us to suggest creative solutions to the uncertainty and lack of information about the intended use of the models in the case studies, which will be useful in ensuring that the models will be flexible and applicable to a wide range of scenarios. Continuous interaction with the case studies, flexible solutions to accommodate for changing requirements and simulations were adopted in order to allow us to achieve the first-year goals.

7. Instructions (locations of material)

This document can be found at ZENODO (DOI according to version) 10.5281/zenodo.4708753

The source code Version 1.0 for the Social Recommender can be found at ZENODO 10.5281/zenodo.4724887. Latest version is 10.5281/zenodo.4724886.

A draft version of the social REST API can also be found at:

https://app.swaggerhub.com/apis/ajwecker/SPICE-UserModel-API/v0#/user-controller

8. References

1. Adler, A.C. (1964). Social Interest a Challenge to Mankind.
2. Badami, M., & Nasraoui, O. (2021). PaRIS: Polarization-aware Recommender Interactive System. In Proceedings of the 2nd Workshop on Online Misinformation-and Harm-Aware Recommender Systems (OHARS 2021), Amsterdam, Netherlands.
3. Bruni, L. E., Daga, E., Damiano, R., Diaz, L., Kuflik, T., Lieto, A., . . . Wecker, A. (2020). Towards Advanced Interfaces for Citizen Curation. Proceedings of *AVI2CH* 2020: Workshop on Advanced Visual Interfaces and Interactions in Cultural Heritage (*AVI2CH* 2020). New York: ACM.
4. Ellwood, E., Estes-Smargiassi, K., Graham, N., Takeuchi, G., Hendy, A., Porter, M., & Lindsey, E. (2018). Project Paleo: Citizen Curation and Community Science at the Natural History Museum of Los Angeles County. Biodiversity Information Science and Standards.
5. Kotkov, D., Veijalainen, J., &Wang, S. (2020). How does serendipity affect diversity in recommender systems? A serendipity-oriented greedy algorithm. Computing, 102(2), 393–

[4] https://github.com/axios/axios

411. https://doi.org/10.1007/s00607-018-0687-5

6. O'Neill, R. (2017). The Rise of the Citizen Curator: Participation as Curation on the Web (Doctoral dissertation, University of Hull).
7. Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: introduction and challenges. In Recommender systems handbook (pp. 1-34). Springer, Boston, MA.
8. Robinson, J., & Carletti, L. (2019). Our Theatre Royal Nottingham: co-creation and co-curation of a digital performance collection with citizen scholars. International Journal of Performance Arts and Digital Media, 15(2), 128-148.

9. Appendix

9.1 Recommender File Structure

The structure of the file system is the same as usermodel D3.3 (with files in controller, model and repository directories)

9.2 SPICE Social Recommender API REST

A more detailed description of the APIs can be seen in D6.4

9.3 React example of wrapped REST calls

9.3.1 Recommender Service

```
import axios from 'axios';
const USER_API_BASE_URL = "http://localhost:8080/api/v2/config";
//
class RecommenderService {
    getConfigurations(user,subject,recConfig){
        return axios.get(USER_API_BASE_URL);
    }

}
export default new RecommenderService()
```