

18.10.2016/ak.

SALSAH 2

reDesign of the research platform:

**UX/UI notes and wireframes about the workspace, the
resource objects, the tools and functions**

Contents

1 SALSAH technical aspects	4
2 Simple workflow in a research project	4
3 SALSAH architecture	6
3.1 salsahFramework	6
3.2 salsahSearch	8
3.2.1 simpleSearch	8
3.2.2 extendedSearch	9
3.2.3 facetedSearch	10
3.3 salsahView	11
3.3.1 gridView (search results and collections)	11
3.3.2 listView (search results)	11
3.3.3 tableView (search results of one resource type)	12
3.3.4 resourceView	13
3.3.5 splitView (compare resources)	13
3.3.6 graphView	13
3.3.7 dashboardView	13
3.4 salsahObject	15
3.4.1 emptyObject	15
3.4.2 imageObject	15
3.4.3 documentObject	15
3.4.4 videoObject	16
3.4.5 audioObject	16
3.4.6 collectionObject (collection / link / book) / salsahCollection . . .	16
3.4.7 regionObject (for images and documents)	17
3.4.8 sequenceObject (for video and audio)	17
3.5 salsahProperty	18
3.5.1 stringElement (text)	18
3.5.2 numberElement (integer or floating point)	18

3.5.3	richtextElement (textarea)	18
3.5.4	dateElement (incl. period)	18
3.5.5	timeElement (incl. interval)	18
3.5.6	locationElement (connection to geonames.org)	18
3.5.7	resourcePointerElement (autocomplete or dropdown)	18
3.5.8	selectionElement (dropdown, checkbox or radio)	18
3.5.9	hierarchicalListElement (dropdown or radio)	18
3.6	salsahSettings	19
3.6.1	projectSettings	19
3.6.2	userSettings	19
3.7	salsahExchange	20
3.7.1	importTools	20
3.7.2	exportTools	20
3.7.3	Some notes about the import (and export) process	20

1 SALSAH technical aspects

The idea of SALSAH – as a virtual research platform – is to help researchers to handle their project, to organize a team and to work with digital sources. At the end it should be possible to present the data and the results on an own web publication platform.

SALSAH uses the restFul API of Knora (knora.org) – a software framework for storing, sharing, and working with primary sources and data in the humanities. Images will be handled by the Sipi server; multimedia stuff like video and audio files we should use FFserver and FFmpeg.

This document describes SALSAH as the generic front end (GUI) of the whole system. SALSAH will be build by the Angular2 framework (angular.io) and styled with google's material design (material.angular.io).

2 Simple workflow in a research project

A research project starts with an idea and the research question. It's the beginning of the conceptual phase where the researcher has to define his sources and methods and where he forms a team. The main work is the second part, where the team works with the data and the sources: Collecting, organizing and evaluating. At the end of the project stands the publication of the data and results.



Figure 1: Research project workflow (from left to right) in an analogous way.

SALSAH is based on a similar workflow concept. The version 1 (salsah.org (18.10.2016)) has an administrator area (salsah.org/admin) where the user can define his project and sources. The team works on the main URL salsah.org. The path /projectname is reserved for the project presentation/publication webpage.



Figure 2: SALSAH 1 with admin area (concept), work space (research) and project page (publication).

For SALSAH version 2 we're planning a different structure: The admin and the workspace interface would be on the same app route. The access to the admin area (projectSettings) depends on the user's rights. The project publication page can reuse the existing modules from SALSAH. But at the moment it's not clear how the URL structure will look like. ...
[http://dasch.org/\[project\]/\[uuid\]](http://dasch.org/[project]/[uuid]) or [http://project.knora.org/\[uuid\]](http://project.knora.org/[uuid])

The Data and Service Center for the Humanities (DaSCH) will use the SALSAH user interface. The DaSCH is implementing a distributed architecture with local or domain specific nodes which form a distributed peer-to-peer network. Resources can be made available to the other nodes at the discretion of each individual node. Authentication and user access is also organised in a distributed manner where each node has control whom it "trusts". In the near future, the Switch "Swiss edu-ID" will be used. For more information about Knora and the DaSCH: read the paper by Lukas Rosenthaler: Data and Service Center for the Humanities – A National Service Architecture, July 2016.

3 SALSAH architecture

3.1 salsahFramework

The main SALSAH app component is the framework module which builds the default layout template with

- salsahHeader (Fig. 5) with
 - project selection and settings menu
 - search bar panel incl. simple and extended search
 - import menu (add resources and create collections)
 - documentation menu (link to the documentation and/or cheat sheet)
 - user menu (profile setting, log out etc.)
- salsahFacetedSearch on the left hand side
- salsahView as a main container for salsahSearchResults, the graph and resource viewer, dashboard etc. The salsahView would be the frame for the ng2 router-outlet.
- salsahFooter (?)



Figure 3: salsahFramework

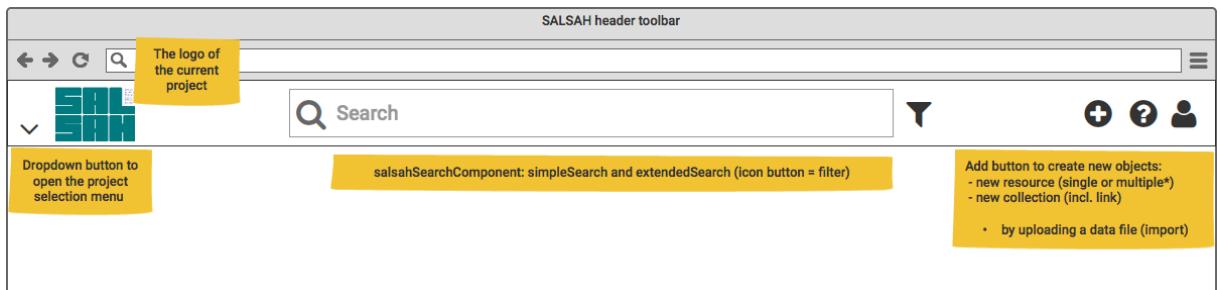


Figure 4: salsahHeader

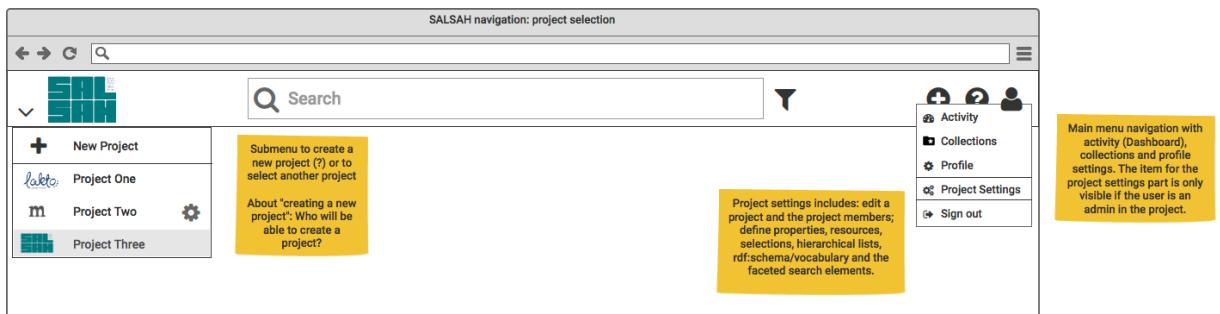


Figure 5: salsahHeader

3.2 salsahSearch

3.2.1 simpleSearch

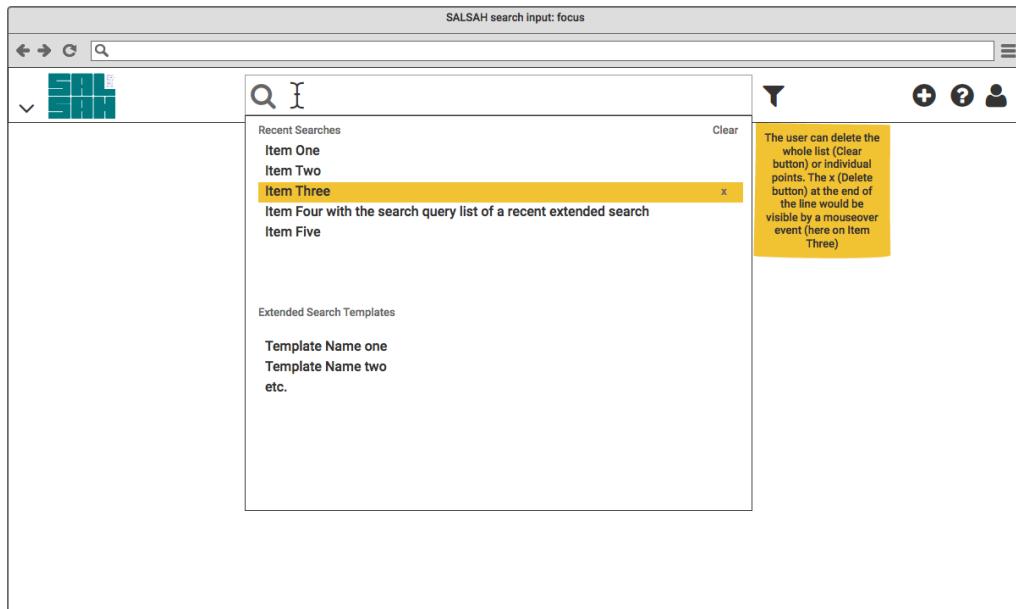


Figure 6: salsahSearch: focus on simpleSearch input shows recent search queries and saved extended search templates

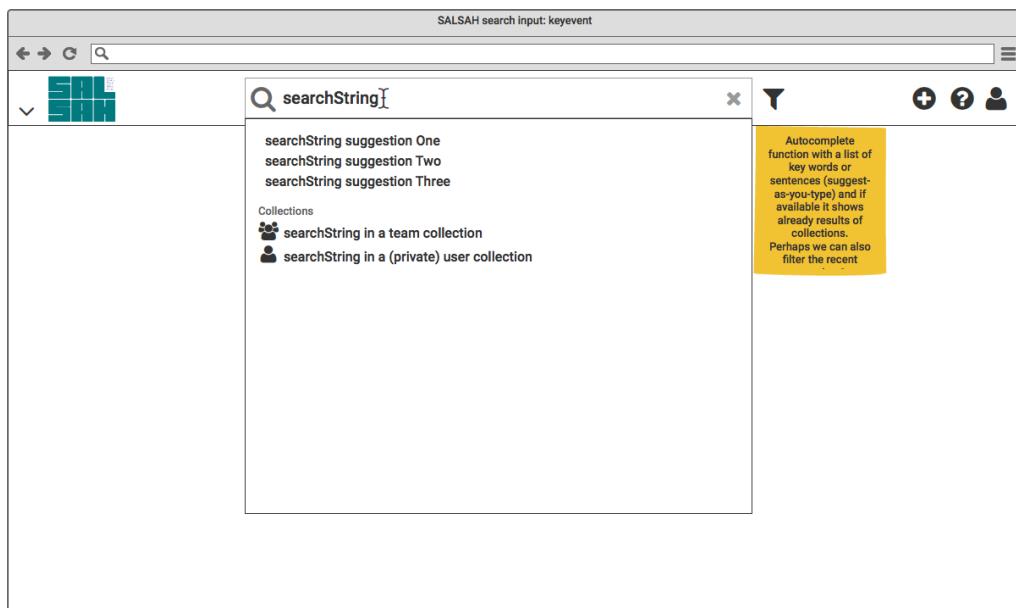


Figure 7: salsahSearch: keyevent in simpleSearch input shows a suggest-as-you-type list (incl. collection suggestions)

3.2.2 extendedSearch

The extended search could look like a filter setting page in an e-mail application.

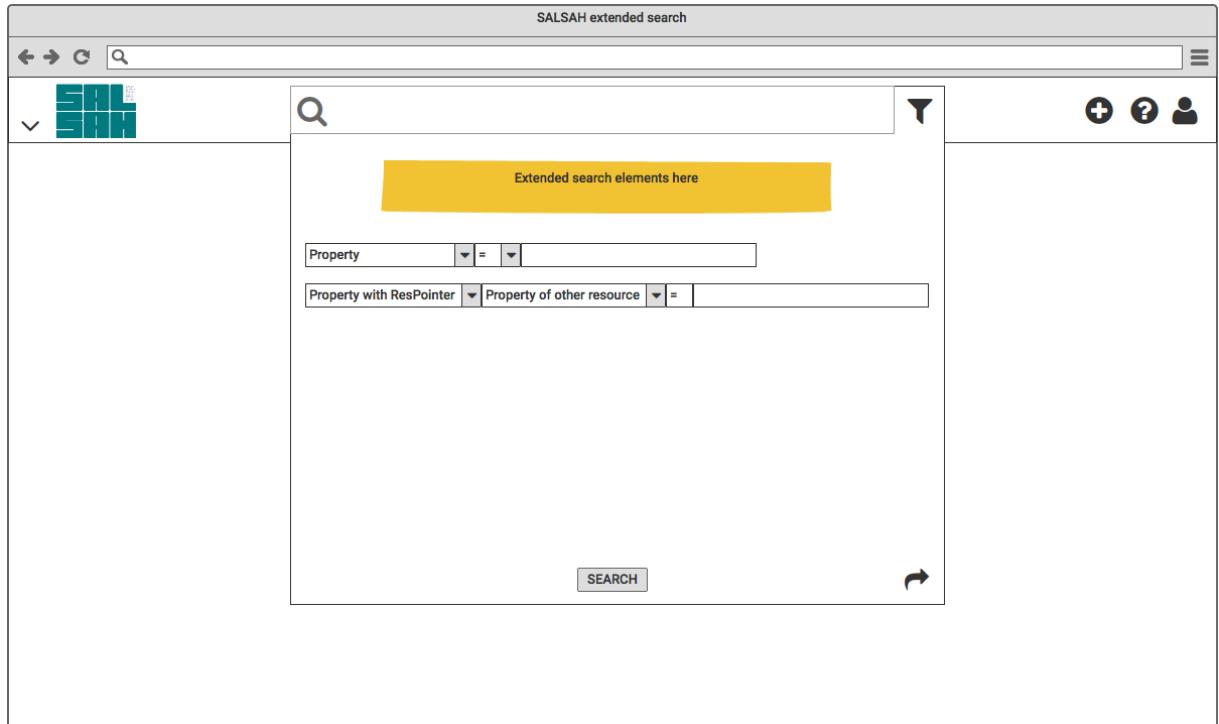


Figure 8: salsahSearch: extended search box

3.2.3 facetedSearch

The screenshot shows a web-based search interface titled "SALSAH faceted search". At the top, there is a search bar with the placeholder "search query : (incl. extended search parameters)". To the right of the search bar are several icons: a magnifying glass, a plus sign, a question mark, and a user profile. Below the search bar, a yellow sidebar contains the text: "Facetted search here. The facetted search elements are defined in the projectsSettings area for the salsahResource." The main content area is titled "Results 12345". It displays four search results, each consisting of a thumbnail image (a white square with a black 'X'), a preview text ("just a few properties here as a preview. click on the element to open it"), and a timestamp ("Last changes on dd. mm. yyyy hh:mm by user"). Below the results are three small icons: a gear, a document, and a right-pointing arrow, followed by the text "graph collection share/save". At the bottom of the page is a navigation bar with icons for back, forward, and search.

Figure 9: salsahSearch: faceted search box on the left hand side

3.3 salsahView

3.3.1 gridView (search results and collections)

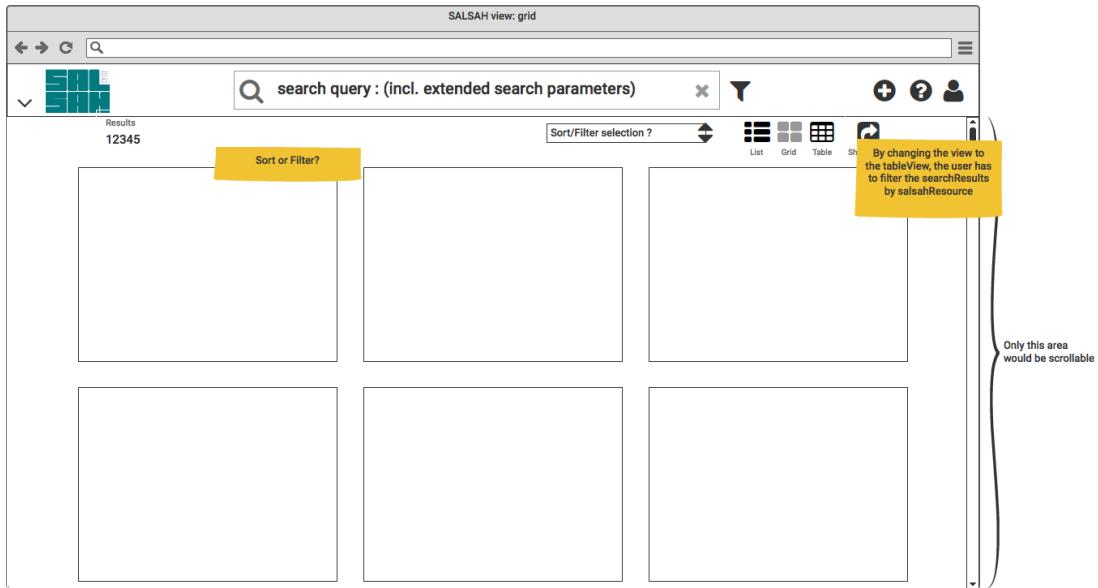


Figure 10: salsahView: grid (inspired by the old light table and the LFI news)

3.3.2 listView (search results)

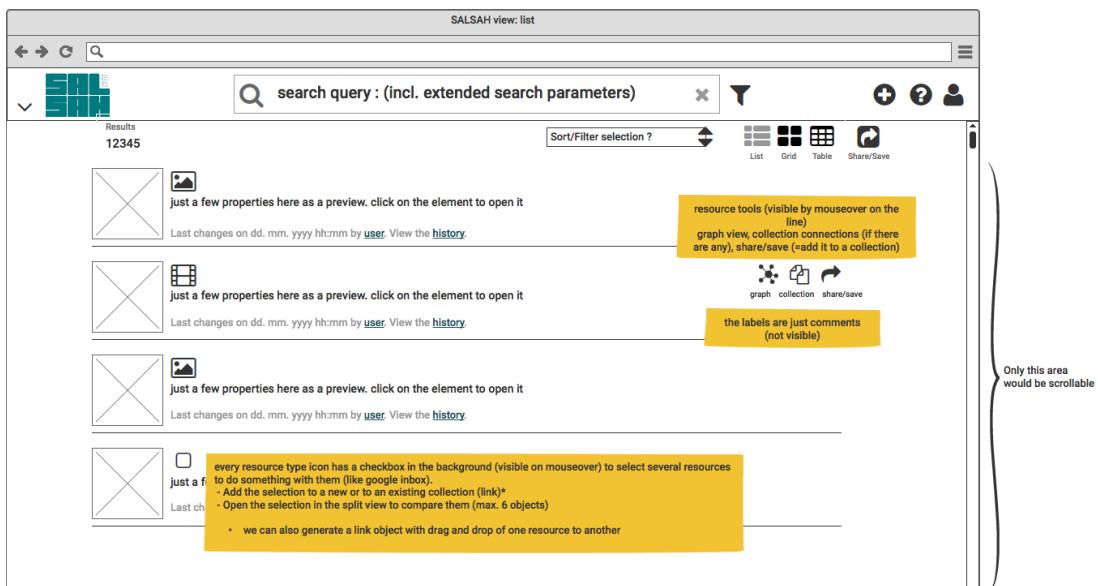


Figure 11: salsahView: list (inspired by Google's inbox)

3.3.3 tableView (search results of one resource type)

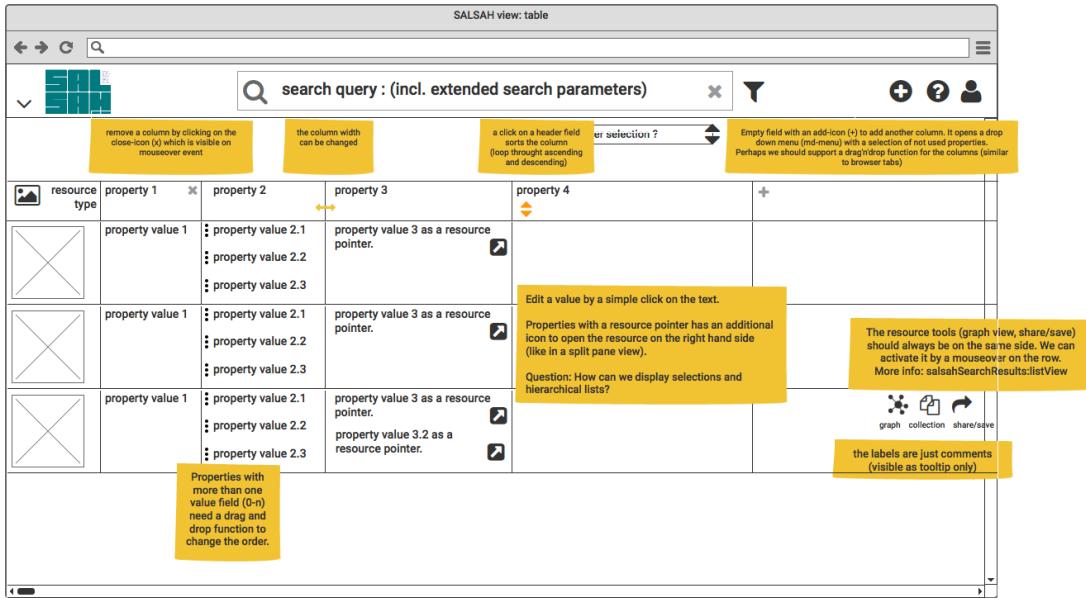


Figure 12: salsahView: table (inspired by Excel)

From the list view (grid, list or tableView) the user can select more than one resource (well known from e-mail applications). In that case, an additional header provides some tools to do something with the selected resources. If the user selects some resources which are from the same resource type, he can edit them by one click. A modal box shows the properties of the selected resources. It's the same process as Apple's iTunes has (s. Fig. 13).

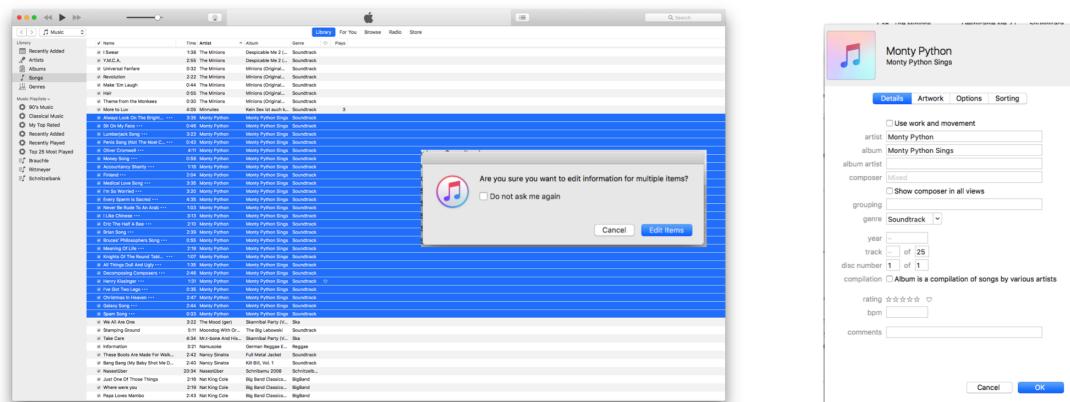


Figure 13: Example from iTunes: Selection of some songs; “Get info” opens a modal box, where the user can edit the properties for the whole selection.

3.3.4 resourceView

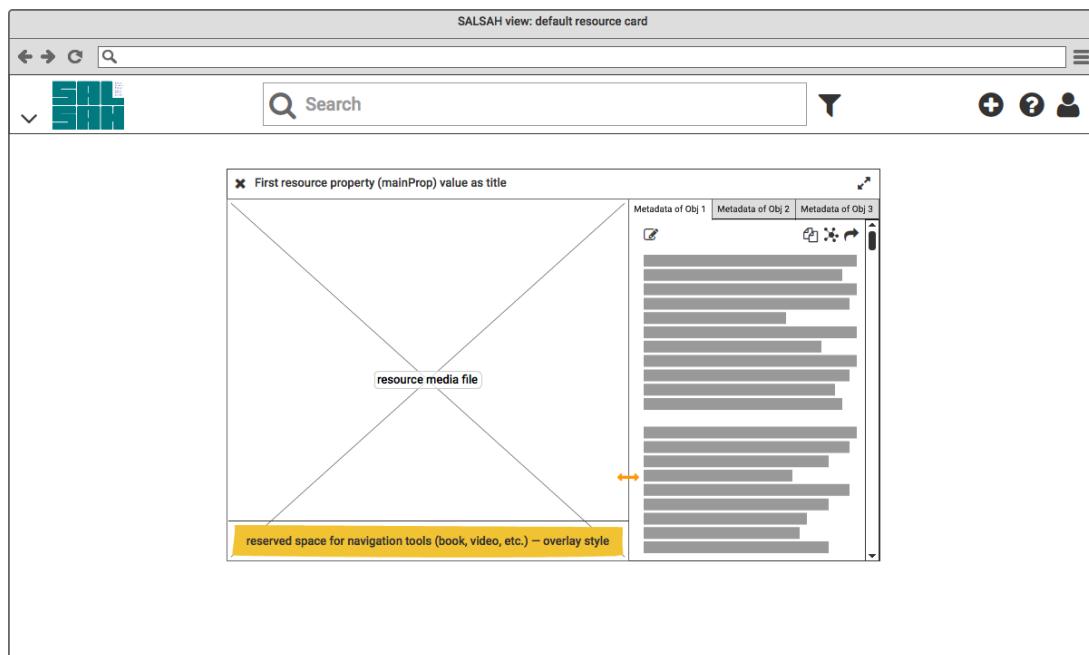


Figure 14: salsahView: default resource card/modal (inspired by the window element of elementary OS)

3.3.5 splitView (compare resources)

Resource comparison viewer with a maximum of six flexible boxes.

3.3.6 graphView

Something with D3.js (d3js.org)

3.3.7 dashboardView

Place for activity thread includes updates from all projects the user is part of or has subscribed to. Activity thread should also be possible for user's activity.

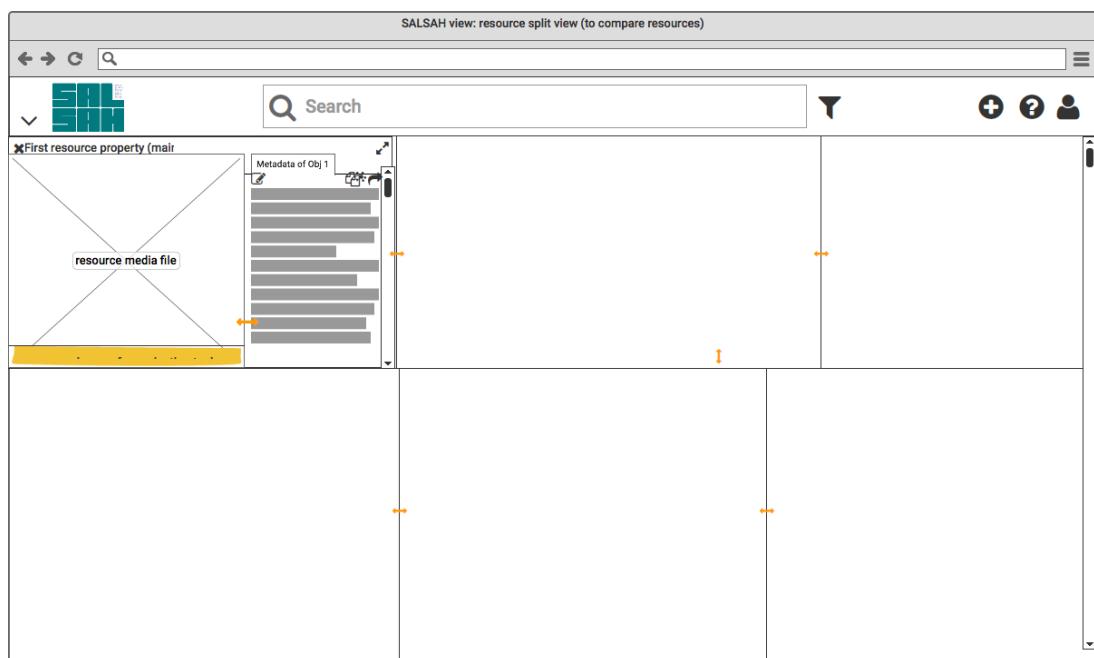


Figure 15: salsaView: split (inspired by codepen.io)

3.4 salsahObject

Like in SALSAH 1 we have a few predefined default resource objects. Most of the objects depend on the different kind of media types: emptyObject (object without a file, metadata only), imageObject, documentObject, videoObject, audioObject, collectionObject, regionObject, sequenceObject. Every salsah resource object needs his own viewer environment (card) with specific predefined tools.

- **close** and **resize** the resource card; fullscreen and back to card size; (minimize? add the resource to one field (of 6) in the split view.)
- **change the view:** switch to the graph viewer or to the collection object, if the object is stored in a collection (link object).
- **share/add:** share the resource with friends or use the URI somewhere else (depends on the resource rights) / add the resource to a collection package (sth. like a playlist in music apps) – we're using the collection also for links between at least two resources.
- **edit** (incl. delete) the resource (depends on the resource and property rights).

3.4.1 emptyObject

The empty object is an object without a media file; it has only metadata and the default tools as described above.

3.4.2 imageObject

The image object has additional tools like: zoom, quality changer, rotate, mirror, transcriber, regions marker

3.4.3 documentObject

The document object is for pdf, latex or rtf, but also word etc. We're not able to display all the different document types – in that case, we offer a download button. The additional tools depending on the document viewer. But we need (+/-) zoom, quality changer, rotate, mirror, transcriber, regions marker.

3.4.4 videoObject

The moving image needs some more tools: timeline with preview, navigation (play, pause, stop, forward and rewind), scroll through, change quality, sequence marker (start/end), frame extraction, transcriber (incl. musical notation), transcription import (subtitle files e.g. srt)

Inspired by existing tools: Transcribe, f4, ELAN, MaxQData

Seq	Title	Camera	Description	Dialogue	Sound
00:58:10:10 01:00:16:07	In the Darkroom	Full Shot, 3/4 Shot, Medium	The photographer develops the film and makes blow-ups.		
01:00:16:07 01:01:44:00	The blow-ups	Full Shot, Medium	The photographer pins the photos up on the walls. He's juggling their position to create some kind of order.	We can reuse the salsahView:table for the salsahSequenceProtocol	
01:01:44:00 01:01:46:07	Photo detail: Couple	close up	One of the photo in a detail view (2s). Shot of the couple by trees embracing, with man's back to the camera.		
01:01:46:07 01:02:54:23	The detective story begins	Medium close up, Medium	MCU of Photographer from in back of the photos through the gap between them. Puts on a record, gets whiskey. Camera shows him studying the photos, drinking.		Music by Herbie Hancock starts at 01:02:18:04
01:05:38:20 01:06:17:13	Phone call	3/4 Shot	Photo in the background: Camels	Photographer: - Hello? - Knightsbridge 1 - 239. what? No, I'm sorry.	

Annotations:

- Top left: Camera tilts up to show the photographer studying the middle photo (#2) with magnifying glass. He looks where the woman is looking. He looks where the woman is looking. Discovers something and marks a rectangle on the area of the fence she's looking at.
- Bottom left: Radio buttons for 'reuse sequence' or 'new sequence' and a 'SAVE' button.

Figure 16: video Object (top left) embedded in the sequenceTranscription tool

3.4.5 audioObject

The audioObject could be similar to the videoObject. Perhaps we need an extra object for musical notes (musicNotationObject ?)

3.4.6 collectionObject (collection / link / book) / salsahCollection

collectionObject/linkObject

A collection connects various salsahObjects (resources). We can reuse the salsahCollection component for links and resource annotations as well. Every user should be able to create

a collection and he can share it with others (share with the whole project team or with single user).

bookObject The book is a special collectionObject with a navigation tool (sth. like timeline) incl. page preview and flick through. The single page includes the tools from the imageObject.

3.4.7 regionObject (for images and documents)

3.4.8 sequenceObject (for video and audio)

3.5 salsahProperty

various GUI elements for different properties like salsahSelection (simple and hierarchical), salsahLocation (geoname connection), salsahResourcePointer, salsahText, salsahNumber, salsahDate (simple and as a period), salsahTime, salsahRegion (?)

3.5.1 stringElement (text)

3.5.2 numberElement (integer or floating point)

3.5.3 richtextElement (textarea)

3.5.4 dateElement (incl. period)

3.5.5 timeElement (incl. interval)

3.5.6 locationElement (connection to geonames.org)

3.5.7 resourcePointerElement (autocomplete or dropdown)

3.5.8 selectionElement (dropdown, checkbox or radio)

3.5.9 hierarchicalListElement (dropdown or radio)

3.6 salsahSettings

3.6.1 projectSettings

Settings area for project admins.

projectSettings includes:

projectProfile, projectMembers, projectVocabulary, projectProperties, projectSelections, projectHierarchicalLists, projectResources, projectSearchTemplates / projectFacetedSearch

3.6.2 userSettings

Settings area for the user's profile.

3.7 salsahExchange

3.7.1 importTools

The SALSAH module “importTools” includes all kind of tools to add new objects into a project repository. They could be: `createNewResource` (single file upload (*or addNewResource ?*)), `addNewResources` (multiple file upload¹), but also `createNewCollection`, `createNewLink` and `createNewProject` (here we’re not sure yet).

We’re using the importTools with the top-level menu button “add” (+) – selection menu – and a modal box.

3.7.2 exportTools

Export a dataset as CSV...

3.7.3 Some notes about the import (and export) process

Three possible cases to transfer or to create data

It’s important to understand, at least to some degree, the subject or research of each project. This is necessary in order to translate (in the case of post mortem or *in vivo*) or create (*ab ovo*) an adequate data model to represent the data in the platform. The task of creating a data model requires considerable direct interaction with the researchers.

In the case of planned or starting projects (“*ab ovo*”), one difficulty is that researchers are not always familiar with the important concepts for their data models. Several examples (e.g. the Schweizerische Gesellschaft für Volkskunde, and the Anton Webern Edition) have shown that creating an adequate and efficient data model is essential for the success of a research project.

Already active projects (“*in vivo*”) often use tools and data models that are not optimal for the given task. Migration into the platform is often an opportunity to clean up the project’s data models.

¹In SALSAH v1 we’re using special PHP scripts to upload more than one file per time. These scripts are able to read csv-, filemaker- or other exported files. For SALSAH 2 it should be possible to have a feature like this in the front end. But at the moment it has a low priority status, because it’s awkward to implement.

Post-mortem integration poses the biggest challenge. In one of the major test cases, the Lexicon Iconographicum Mythologiarum Classicarum, there is no documentation available, and the people that created the data models and software are no longer available. However, there are still many active users who can help us to understand the concepts. Still, such projects require a great deal of time-consuming reverse engineering.

The two last cases (“in vivo” and “post mortem”) need special support from our side. Here we have to write an import script depending on the existing data. Perhaps it’s possible to have an import button, where the user can upload the data exported from his previous database (Filemaker, MySQL, etc.).