# xrf-explorerV2

**FILE: XRF_ExplorerV2_userrequirementsdocument.pdf**

**General information on the software documentation**
**version: 2024 October 02**

---

**xrf-explorerV2** is a research software toolkit developed by students at Eindhoven University of Technology (TU/e), for the Van Gogh Museum Amsterdam in a partnership with ASML. The toolkit is intended to be used for the integrated exploration of multimodal images, spectral data, and chemical mappings of a painting. The toolkit is developed to assist in the conservation science practice.

**Project**
The original prototype (V1) was created by a TU/e master's student from January–September 2023. The current version (V2) was developed by a team of TU/e bachelor students from April–July 2024.

**Documentation**
Five documentation files were delivered by the development team.
Together they form a comprehensive documentation package:

- XRF_ExplorerV2_userrequirementsdocument.pdf
- XRF_ExplorerV2_softwaredesigndocument.pdf
- XRF_ExplorerV2_acceptancetestplan.pdf
- XRF_ExplorerV2_softwaretransferdocument.pdf
- XRF_ExplorerV2_softwareusermanual.pdf

**Acknowledgements**
For the development, testing, and documenting of xrf-explorer, test data was provided by the Museum of Modern Art (New York) to the development team. It consisted of images and xrf scanning data of the "Portrait of Joseph Roulin" painting by Vincent van Gogh. Images of this painting are included in some of the documentation files for the purpose of explaining the software usage.

For more detailed information on this painting, including image licenses, we refer to:
https://www.moma.org/collection/works/79105



**Software**
The xrfexplorerV2 software itself (codebase and code documentation) can be found at the github repository "Olive-Groves/xrf-explorer" (see: https://github.com/olive-groves ).

EINDHOVEN UNIVERSITY OF TECHNOLOGY

# User Requirements Document

2IPE0 SOFTWARE ENGINEERING PROJECT

July, 2024

**GROUP 6**


## PROJECT TEAM

| | |
|---|---|
| Adrien Verriele | 1710303 |
| Diego Rivera Garrido | 1674196 |
| Dirk Burgers | 1653873 |
| Iliyan Teofilov | 1671952 |
| Ivan Ivanov | 1661469 |
| Jan Bulthuis | 1696866 |
| Lotte Lakeman | 1668137 |
| Massimo Leal Martel | 1662147 |
| Pablo Benayas Penas | 1667939 |
| Ruben Savelkouls | 1695347 |
| Sonia Maxim | 1675656 |

## PROJECT MANAGERS

Antreas Efstathiou
Konstantinos Chanioglou

## SUPERVISOR

Prof dr. R. Bloo

## CLIENTS

Ana Martins
Lars Maxfield
Marco Roling

# ABSTRACT

This document, the User Requirements Document (URD) serves as a comprehensive guide for the development of the XRF Explorer 2.0. The XRF Explorer 2.0 is a web application aimed at conservation scientists, that facilitates the analysis of paintings composition. This application builds upon the foundations laid by Dominique van Berkum's master thesis project introducing the proof of concept for XRF-XPLORER [1], and it is tailored to fit the requirements of the clients. The document outlines the purpose and scope of the project and provides a detailed list of definitions, to ensure clarity throughout the document. The document's content is structured into multiple sections covering the general description of the product and the specific requirements. Important use cases are presented to illustrate various functionalities and scenarios of the application.

# Contents

# DOCUMENT STATUS SHEET

## General

| | |
|---|---|
| **Document title** | User Requirements Document |
| **Document identifier** | URD/2.2 |
| **Document authors** | Diego Rivera Garrido<br>Dirk Burgers<br>Iliyan Teofilov<br>Lotte Lakeman<br>Massimo Leal Martel<br>Ruben Savelkouls<br>Pablo Benayas Penas<br>Sonia Maxim |
| **Document status** | Final |

## Document history

| Version | Date | Authors | Reason |
|---|---|---|---|
| 0.01 | 24-04-2024 | Lotte Lakeman | Created document from document template |
| 0.02 | 24-04-2024 | Diego Rivera Garrido<br>Iliyan Teofilov<br>Lotte Lakeman<br>Ruben Savelkouls<br>Sonia Maxim | Added first draft of requirements in chapter 3 |
| 0.03 | 25-04-2024 | Lotte Lakeman | Added the initial version of the introduction |
| 0.04 | 25-04-2024 | Lotte Lakeman | Added the initial version of sections 2.1, 2.3, and 2.4 |
| 0.05 | 25-04-2024 | Diego Rivera Garrido<br>Iliyan Teofilov<br>Ruben Savelkouls | Started categorising requirements more logically |
| 0.06 | 26-04-2024 | Diego Rivera Garrido<br>Iliyan Teofilov<br>Ruben Savelkouls<br>Sonia Maxim | Continued categorising requirements |
| 0.07 | 26-04-2024 | Lotte Lakeman | Added the initial version of the use cases. |
| 0.08 | 27-04-2024 | Sonia Maxim | Added the MoSCoW prioritization. |
| 0.09 | 28-04-2024 | Diego Rivera Garrido | Added the initial version of the term definitions. |
| 0.09 | 28-04-2024 | Diego Rivera Garrido<br>Ruben Savelkouls<br>Iliyan Teofilov | Finished categorising the requirements. |
| 0.10 | 28-04-2024 | Sonia Maxim<br>Lotte Lakeman | Continued writing use cases. |
| 0.11 | 28-04-2024 | Lotte Lakeman | Continued writing sections 2.2 and 2.4. |

Continued on next page

| Version | Date | Authors | Reason |
|---|---|---|---|
| 0.12 | 28-04-2024 | Sonia Maxim | Added abstract. |
| 1.0 | 28-04-2024 | Diego Rivera Garrido<br>Dirk Burgers<br>Iliyan Teofilov<br>Lotte Lakeman<br>Ruben Savelkouls<br>Pablo Benayas Penas<br>Sonia Maxim | Finalized all sections of the document for submission of the draft version. |
| 1.1 | 01-05-2024 | Diego Rivera Garrido<br>Dirk Burgers<br>Iliyan Teofilov<br>Lotte Lakeman<br>Ruben Savelkouls<br>Pablo Benayas Penas<br>Sonia Maxim | Changes according to the client feedback for Sections 1,2 and 3. |
| 1.2 | 02-05-2024 | Diego Rivera Garrido<br>Dirk Burgers<br>Sonia Maxim | More changes according to the client feedback for Sections 1,2 and 3. |
| 1.3 | 03-05-2024 | Diego Rivera Garrido | Changes to use cases to reflect changes done to requirements. |
| 1.4 | 06-05-2024 | Diego Rivera Garrido<br>Dirk Burgers<br>Sonia Maxim<br>Lotte Lakeman | Final revision. |
| 1.5 | 12-05-2024 | Lotte Lakeman | Implemented the clients' feedback. |
| 2.0 | 16-05-2024 | Lotte Lakeman<br>Massimo Leal Martel | Finalized URD. |
| 2.1 | 29-05-2024 | Lotte Lakeman | Cleaned up requirements. |
| 2.2 | 23-06-2024 | Dirk Burgers | Finalized the use cases. |

# DOCUMENT CHANGE RECORDS

| Version | Date | Section | Reason |
|---|---|---|---|
| 0.1 | 24-04-2024 | Entire document | Creation of the entire document |
| 0.2 | 24-04-2024 | Section 3 | First draft of requirements |
| 0.2 | 24-04-2024 | Sections 1, 2 & 3 | Initial versions of introduction and sections 2.1,3,4, and started categorising requirements |
| 0.3 | 26-04-2024 | Appendix A | Started defining use cases |
| 0.4 | 28-04-2024 | Abstract and Sections 1 & 3 | Finished categorising requirements, started the list of definitions and wrote abstract. |
| 1.0 | 02-05-2024 | All sections | Changes according to client |
| 1.1 | 06-05-2024 | All sections | Final changes according to client feedback |
| 2.0 | 16-05-2024 | All sections | Final changes according to client feedback |

# 1 INTRODUCTION

## 1.1 Purpose

This document, the URD (User Requirement Document), serves as a document in which the general description of the application (XRF Explorer 2.0), its specific requirements and various use cases are explored and explained. This document has been created in communication with the clients, Ana Martins, Lars Maxfield, and Marco Roling to together shape the outline of the XRF Explorer 2.0 application. The document will serve as both guidance for development throughout the project, as well as a contract between the development team and the clients. Any changes made to the requirements must be approved by both the developers and the clients.

The proposed requirements have been shaped through conversations with the clients, the base work and research presented as the thesis ”XRF-XPLORER: An interactive visual exploration tool for micro-X-ray fluorescence scanning data on paintings” [1], and exploration in the application XRF-XPLORER [1.0].

## 1.2 Scope

The XRF-XPLORER as created by Dominique van Berkum is a proof of concept for an application which allows conservation scientists to better visualize and understand Micro-XRF data alongside RGB, UV and X-ray images of paintings [1].

The goal of this software engineering project is to turn this proof of concept into a functional web application. The XRF Explorer 2.0 will be a viable and user-friendly web application in which conservation scientists can freely explore collected painting data. Additional attention will be paid towards the documentation of the software such that potential future wishes from conservation scientists can be implemented with ease.

## 1.3 List of definitions

### 1.3.1 Terms

| Term | Description |
|---|---|
| Active view | A view that is active and currently visualizing something. |
| Alphascale | Refers ”to a grayscale image converted such that the transparency of each pixel (the alpha channel) is proportional to how black or white the original grayscale pixel is” [2]. |
| Clients | Ana Martins, Lars Maxfield, and Marco Roling |
| Color coding | Colors chosen to represent or differentiate elements in the application. |
| Color segmentation view | Visualization view that displays the distribution of color segments throughout the painting. |
| Color segment | Area of pixels which all have a similar RGB color. |
| Contextual image | Image displaying the painting in a modality, e.g. RGB images or UV images. |
| Context visualization | Visualization view that displays an RGB zoomed-in image of the area around the lens. |
| Control point | A point in pixel space used to define the coordinates of a distinguished feature in an image. |

Continued on next page

| Term | Description |
|---|---|
| Coordinate system | System to uniquely determine the position of points in space. |
| Data registration | The process of aligning images using (given) control points such that all the images have the same dimension and orientation. |
| Data source | Collection of data and files related to a single painting that are used to analyze said painting. The collection includes raw, processed and contextual data. In addition, it containsrecipes to link the different data files together. Note: in the final software these are referred to as **projects**. |
| Data source list | List of available data sources. |
| Default visual settings | The default values of the visual settings. |
| Dimensionality reduction | Transformation of data from high-dimensional space into low-dimensional space. |
| Dimensionality reduction system | The subsystem that handles the dimensionality reduction logic and visualization. |
| Documentation | The documentation refers to the set of informational resources that provide instructions on the utilization of the application as well as an explanation of the application's source code. |
| Elemental channel view | Visualization view that displays the elemental channels. |
| Elemental composition | Abundance and distribution of the elements present in the painting. |
| Elemental distribution map | Image derived from the processed data, visualizing the distribution of one element across the painting. |
| Elemental distribution view | Main viewer when the top layer of the main viewer is set to the elemental distribution map. |
| Element visualization | Visualization view that displays the elemental composition across the painting as well as the selected area. |
| Elements | Chemical elements present in the painting. |
| Filters | Visual effects applied to a layer visible in the main viewer (includes contrast, saturation, gamma and brightness). |
| Lasso selection tool | Selection tool for the dimensionality reduction view. |
| Layer | Discrete compound that contains an individual contextual image or elemental distribution map. |
| Layer system | Structured arrangement of different layers, which allows for the layers to be moved, removed and added. |
| Layer stack | Collection of all layers within the application, ordered in a sequence that determines their display priority and visual overlap. |
| Layer view | View in which the layer stack can be seen and edited (move, add, remove layers). |
| Layer configuration | Arrangement of layers together with their configuration (opacity, element(s) being displayed, and order). |
| Lens | Mouse position driven ocular that looks through the top layer to a selected layer below. |

Continued on next page

| Term | Description |
|---|---|
| Lens viewing mode | Mode in which the user controls the lens. |
| Processed data | Processed datacube (3-dimensional) of elemental distribution data obtained from processing the raw data that, for each pixel, gives the abundance of the different elements present. |
| Raw data | Raw datacube (3-dimensional) of spectral data obtained from the XRF scanner that, for each pixel, gives the intensity of the X-ray fluorescence emitted at different energies the elements in the painting. |
| Recipe | A set of control points over a set of images of the painting, linked across multiple images for perspective correction and alignment to register the data. |
| Selection bar graph | Visualization view that displays the average elemental composition of the user's current selection. |
| Selection tool | Tool that allows the user to select a subset of the painting. |
| Selected areas | Areas selected by the user using the selection tool in the main viewer. |
| Non-selected areas | Areas not selected by the user using the selection tool in the main viewer. |
| Spectra | Representation of the intensity (counts) of the fluorescence emitted by the materials in the painting as a function of energy (KeV). |
| Spectral visualization | Visualization view of the average fluorescence spectrum acquired over the whole painting or selected area. |
| View | A main viewer or a visualization view. |
| (Main) Viewer | Main large interactive visualization where spatial data is presented. |
| Visual settings | The value of the opacity, focus, colors of highlighted elements. |
| Visualization views | Any of the visualizations that are not the main viewer, where different plots and graphs can be displayed. |
| Open source license | A license that grants permission for anybody to use, modify and share the licensed software for any purpose preserving the openness of the software [3]. |

### 1.3.2 Acronyms and abbreviations

| Term | Description |
|---|---|
| URD | User Requirements Document |
| XRF scanner | X-ray fluorescence scanner |
| DRS | Dimensionality reduction system |
| URC | User Requirements Constraint |
| URF | User Requirements Functional |

## 1.4 List of references

[1] D. V. B. van Berkum, "Xrf-xplorer: An interactive visual exploration tool for micro-x-ray fluorescence scanning data on paintings," Master's thesis, Eindhoven Univerisity of Technology, September 2023.

[2] L. Maxfield, "butterfly_viewer." https://olive-groves.github.io/butterfly_viewer/butterfly_viewer.html.

[3] GitHub, Inc., "Licenses." https://choosealicense.com/licenses/.

[4] H. Chopp, A. McGeachy, M. Alfeld, O. Cossairt, M. Walton, and A. Katsaggelos, "Image processing perspectives of x-ray fluorescence data in cultural heritage sciences," *IEEE BITS the Information Theory Magazine*, vol. 2, no. 1, pp. 20–35, 2022.

[5] A. Popa, "Visualayered: Combined visual analysis of ma-xrf and ris data andra popa," Master's thesis, Delft University of Technology, December 2022.

[6] Adobe, "Layers in photoshop." https://helpx.adobe.com/photoshop/using/layer-basics.html.

[7] M. Vermeulen, K. Smith, K. Eremin, G. Rayner, and M. Walton, "Application of uniform manifold approximation and projection (umap) in spectral imaging of artworks," *Spectrochimica Acta - Part A: Molecular and Biomolecular Spectroscopy*, vol. 252, 5 2021.

[8] A. Nekvinda, "The moscow prioritization method: Streamlining project management." https://hubstaff.com/blog/moscow-prioritization-method/.

## 1.5 Overview

- Chapter 2 contains the general description of the product, including its product perspective, capabilities, constraints, user characteristics and an environment description. Additionally, the assumptions on which the requirements in chapter 2 are based and the dependencies are mentioned.
- Chapter 3 contains the tables in which the capability requirements are stated per category. The chapter also contains tables in which the constraint requirements are stated. Each requirement has a unique identifier (ID), a requirement description and a priority indicator.
- Appendix A contains the most important use cases and their description. For each use case a title, a summary, preconditions, postconditions, actors, a potential trigger, a priority, a goal, and its related requirements are given. A main flow and potential alternative flow is also stated for each use case.
- Appendix B is the signing page, where the signatures and approval of the client and supervisor can be found.

# 2 GENERAL DESCRIPTION

## 2.1 Product perspective

XRF mapping was developed by cultural heritage scientists and has gained popularity since the late 2000s due to its increased accessibility to scientists. This technique has been used to identify pigments and other materials in order to elucidate the artists' use of materials, to understand how a painted surface has altered over time and to identify anachronistic uses of materials. Such insights are essential for detecting potential fakes, forgeries or past restoration interventions [4]. The analysis method that this project will focus on is Micro-XRF (micro-X-ray fluorescence) scanning. There exist, however, other methods such as X-ray imaging, UV light examination, and high-resolution imaging all of which highlight different areas of the painting [1]. Despite being one of the most powerful forms of painting analysis, there currently does not exist a visualization software for Micro-XRF data which is easily accessible to conservation scientists and addresses all tasks from the user. Similar tools, such as the VisuaLayered [5], have been developed as part of other master's theses, however they are not readily available for conservation scientists.

To resolve this issue, a proof of concept was finalized in 2023 in collaboration between Eindhoven University of Technology, the Van Gogh Museum and ASML. This proof of concept, the Xrf-xplorer, was developed by Dominique van Berkum for a Data Science and Artificial Intelligence Master's project [1]. To transform this proof of concept into a viable web application, a collaboration between the Technical University of Eindhoven, the Van Gogh Museum and ASML was again established. The intended goal is to create the XRF Explorer 2.0, an accessible browser-based application tool where conservation scientists can view different types of painting data and contextual images in one central place. Unlike previous applications, which treated data sources in isolation, the XRF Explorer 2.0 will combine the strengths of all individual data sources and contextual images and allow for a streamlined process in which the user can view the multidimensional data through various methods such as the lens and dimensionality reduction while maintaining an understanding of the spatial context.

## 2.2 General capabilities

### 2.2.1 Managing data sources

The XRF Explorer 2.0 shall allow users various capabilities concerning the managing of data sources. Users shall be able to add new data sources and edit or delete existing data sources.

#### Explore paintings in the main viewer

Perhaps the most important component is the application's large main viewer. In this main viewer, the user can freely pan around their uploaded painting and zoom in on details. Additionally, the user can enable a second main viewer to compare different areas of the painting side-by-side.

### 2.2.2 Export images

Conservation scientists may want to use the images of paintings or charts they have created. To accompany this need, the XRF Explorer 2.0 shall support the ability to export both images of paintings as well as charts generated by the user.

### 2.2.3 Use the layer system

The XRF Explorer 2.0's main advantage is the ability to view a painting's data and contextual images in one, easy-to-use, application. For this, a layer system will be available, similar to that of applications like Photoshop [6]. In the layers panel, the user has the ability to reorder the layers and change the opacity of specific layers (to which the main viewer will be updated accordingly).

### 2.2.4 Use the lens

To aid the analysing process, a lens is available to the user. The lens will allow users to look at a layer different from the current top layer, allowing for direct comparison between the two respective layers. Additionally, the lens can be locked in place, allowing for even further painting exploration.

### 2.2.5 Use the selection tool

The XRF Explorer 2.0 features a def:selToolselection tool to better explore an area's data. By selecting a subarea of the painting, the user is able to see a bar chart of the elemental composition of the selected region as well as a spectrum chart. The spectrum chart will additionally also display the average data of the entire painting, allowing conservation scientists to easily explore the difference in spectra between the selected region and the complete painting.

### 2.2.6 Change the contrast and saturation of the image

A standard feature of image visualization programs is the ability to change the contrast and saturation of images. Changing the image's contrast will change the variation in brightness between objects. This allows for additional depth perception of the image, as well as aiding conservation scientists in being able to distinguish details. Additionally, the XRF Explorer 2.0 allows the user to change the saturation of images. By decreasing the saturation the image's colors are more muted, reducing distractions and allowing for easier detection of shapes and textures. Increasing the saturation makes the painting more vibrant, highlighting the details of an image.

### 2.2.7 Use the context visualization

To aid the analysis process, a context visualization is available to the user. It displays the high-quality zoomed-in RGB image around the lens' position, allowing the user to compare the layer displayed by the lens with the RGB image easily.

### 2.2.8 Select elements

A crucial step in analysing a painting is being able to explore its elemental composition. To aid the user in this process, the XRF Explorer 2.0 has various capabilities.

1. **Highlighting an element:** The XRF Explorer 2.0 allows users to select elements they want to explore and the web application will highlight the pixels in which they are present within the painting.
2. **Changing the intensity of an element:** To further analyse the elemental composition of a painting, the XRF Explorer 2.0 supports the option to manipulate the intensity for elements. Changing the intensity of an element changes the brightness of the pixels associated with this element. This feature allows conservation scientists to better visualise the presence of certain elements.
3. **Changing the highlighting color for an element:** In addition to changing the intensity/threshold, the user can also specify the highlighting color to further aid the visualisation process by clearly being able to distinguish separately highlighted elements.

### 2.2.9 Use the color segmentation view

The XRF Explorer 2.0 will have a color segmentation view in which the user can explore the correlation between color of the painting and elemental composition. Similar colors will be grouped together and the user can select a group from the color segmentation view, highlighting all areas of the painting which have been mapped to this color. To allow for an even more in-depth color exploration, the user is also able to select the colors per element.

#### 2.2.10 Use dimensionality reduction

An insightful method of analysing a painting is through dimensionality reduction. The XRF Explorer 2.0 shall support this type of data exploration. Through dimensionality reduction, the user has a clear overview of the elemental similarities and differences in composition across specific regions of a painting. The default parameters chosen for the dimensionality reduction method are based on the recommendation given in the paper [7].

#### 2.2.11 Visualise data in graphs

As an additional method of visualizing the painting's data, the application shall support various data graphs. The spectral visualization shall display the spectra of the current selection and the element visualization shall display the elemental composition of the current selection and selected elements.

### 2.3 General constraints

#### 2.3.1 Operating systems

It was specified by the clients that, regardless of operating system, the user should be able to use the application as intended. Specifically, the clients requested the application to be browser-based, rather than to be installed.

#### 2.3.2 Hardware constraints

Upon consulting the clients regarding any potential hardware constraints, the main issue identified would be the restrictive amount of RAM. After further consulting, it was revealed that the web application should be able to run on 16GB of RAM. The clients mentioned that the initial loading of the files is allowed to take up to three minutes, if the application then runs smoothly while doing the actual exploration. This RAM constraint is something which must be considered while development takes place.

### 2.4 User characteristics

The XRF Explorer 2.0 will not have any user roles as there is only one type of user. Hence, the application won't feature any role-specific permissions. The permissions and capabilities of the user are identical to the mentioned general capabilities.
The decision not to add user roles was based on the initial conversation with the clients. It was clear from this conversation that the focus of the application should be on the painting exploration process. Adding different user roles, a login system and a database to keep track of all these user accounts was not desired and was outside of the scope of this project. Due to the nature of the application, an application which is for a very specific and niche user group, there is a web of trust between the users and the user pool would be very small. For future versions of the application, it would be possible to create different user roles relatively easily. By creating user accounts, data sources could be linked to specific users allowing users to only explore their own data sources.

### 2.5 Environment description

XRF Explorer 2.0 will be designed as a web application, due to the client's preference for avoiding executable installations and the requirement for compatibility across various operating systems.

**Backend:** The backend will utilize Flask, a lightweight web framework written in Python. Flask was chosen for its ability to easily integrate with Python's extensive libraries and because it will allow us to reuse some of the solutions used in the previous version of the application.

**Frontend:** The user interface will be built using Vue.js because of our team's familiarity with the framework, and the D3.js library will be used for creating the data visualizations. Furthermore, considering the application's extensive requirements for image processing and visualization, and the client's emphasis on performance, we have chosen WebGL2 for rendering the interactive images.

**Data Storage:** The host server's local file system will have folders where input files are collected per data source. Files may be loaded into temporary storage and processed by the web application at runtime. Cache and temp storage must be garbage collected after the web application closes to prevent unnecessary storage use and machine load.

## 2.6 Assumptions and dependencies

### 2.6.1 Assumptions

- The user uses one of the following browsers: Chrome (version 124 or higher) or Firefox (version 125 or higher);
- The user uses one of the following operating systems based on x86-64 based architectures: Windows 10, Windows 11 or Ubuntu Server LTS 24.04;
- The user is proficient in the English language;
- The server is functional and accessible while using the application;
- The user operates the application as intended and specified in Appendix A.

### 2.6.2 Dependencies

- Ubuntu server provided by the clients.

# 3 SPECIFIC REQUIREMENTS

This chapter presents the specific requirements the XRF Explorer 2.0 will comply with, prioritised using the MoSCoW [8] technique:

| Priority | Abbreviation | Description |
|---|---|---|
| Must have | M | These are the requirements that are necessary for the successful completion of the project; they are considered mandatory and must be completed. |
| Should have | S | These requirements are still important but not critical to completing the project. They should be completed, as it enhances the final product, but they are optional. |
| Could have | C | These types of tasks are desirable add-ons to the project. They would be nice to have but are not needed for the final product's functionality. These could be included, but only if time and resources permit. |
| Won't have | W | These requirements are not essential and can be excluded from the current delivery cycle. They won't be included in the project. |

## 3.1 Capability requirements

In this section, the capability requirements are specified. Each requirement has a unique ID, starting with URF (User Requirements Functional), indicating that these are the required functionalities of the application.

### 3.1.1 User roles and accounts

Table 1: Requirements for user roles and accounts

| ID | Requirement | Priority |
|---|---|---|
| URF 1.1 | The application shall have user roles. | **W** |
| URF 1.2 | When a user has created a data source, the application shall allow only that user to access that data source. | **W** |
| URF 1.3 | The application shall have different permissions for different users. | **W** |
| URF 1.4 | The application shall allow users to create an account. | **W** |

### 3.1.2 Loading data

Table 2: Requirements for loading data

| ID | Requirement | Priority |
|---|---|---|
| URF 2.1 | The application shall load file format ".RAW" together with ".RPL" for the raw data and the associated metadata. | **M** |
| URF 2.2 | The application shall load file format ".CSV" and ".DMS" for the processed data. | **M** |
| URF 2.3 | The application shall load file formats ".TIFF", ".JPG", ".BMP" and ".PNG" for the contextual images. | **M** |

Continued on next page

Table 2: Requirements for loading data (Continued)

| ID | Requirement | Priority |
|---|---|---|
| URF 2.4 | The application shall load files in the ".CSV" format as generated by Butterfly-Registrator [2] to use as a recipe for registering the data. | **S** |

Table 3: Requirements for data sources

| ID | Requirement | Priority |
|---|---|---|
| URF 3.1 | The application shall provide a means for the user to upload data sources. | **M** |
| URF 3.2 | The application shall allow users to give data sources a name. | **M** |
| URF 3.3 | The application shall require that the user uploads at least one data file. | **M** |
| URF 3.4 | The application shall provide a means of viewing all available data sources. | **M** |
| URF 3.5 | The application shall require data sources to have a unique name. | **M** |
| URF 3.6 | When loading in a data source, the application shall indicate to the user that the data source is being loaded in. | **S** |
| URF 3.7 | When the user leaves a mandatory field empty during the data source uploading process, the application shall notify the user that the fields must be non-empty. | **S** |
| URF 3.8 | When the user uploads a non-compatible file type during the data source uploading process, the application shall notify the user that the file type is non-compatible. | **S** |
| URF 3.9 | The application shall allow users to delete existing data sources. | **S** |
| URF 3.10 | The application shall allow users to enable/ disable elemental maps in existing data sources. | **C** |
| URF 3.11 | The application shall allow users to rename existing data sources. | **C** |
| URF 3.12 | The application shall allow users to upload new data to existing data sources. | **C** |
| URF 3.13 | The application shall have a specific list for recently accessed data sources. | **C** |

### 3.1.3 Processing data

Table 4: Requirements for processing data

| ID | Requirement | Priority |
|---|---|---|
| URF 4.1 | When loading contextual spatial data and given a recipe, the application shall automatically register the contextual spatial data to a coordinate system. | **M** |
| URF 4.2 | When the processed data is given in multiple sections, the application shall stitch the sections together into a single figure. | **C** |

### 3.1.4 Layer system

Table 5: Requirements for the layer system

| ID | Requirement | Priority |
|---|---|---|
| URF 5.1 | The application shall have a layer system. | **M** |
| URF 5.2 | The layer system shall allow the user to reorder the layer hierarchy. | **M** |
| URF 5.3 | The layer system shall have the option to change the opacity of a layer. | **M** |
| URF 5.4 | The main viewer shall display all enabled layers with the specified opacity in the specified order. | **M** |
| URF 5.5 | The layer system shall allow users to enable/disable a layer. | **M** |
| URF 5.6 | When a layer is disabled, the main viewer shall hide that layer. | **M** |
| URF 5.7 | The application shall allow users to delete a layer. | **C** |
| URF 5.8 | The application shall allow users to add a layer. | **C** |
| URF 5.9 | The application shall allow users to rename layers. | **C** |
| URF 5.10 | The layer system shall have the option to reset the opacity of a layer to its original value. | **C** |
| URF 5.11 | When a layer is deleted, the application shall delete the layer from the data source. | **W** |

### 3.1.5 Viewing data

The following requirements relate to the main viewer.

Table 6: Requirements for viewing data

| ID | Requirement | Priority |
|---|---|---|
| URF 6.1 | The application shall allow the user to visualize the contextual images. | **M** |
| URF 6.2 | The application shall provide a means to visualize the raw data as spectra. | **M** |
| URF 6.3 | The application shall allow the user to zoom in and out in the visualization of the main viewer. | **M** |
| URF 6.4 | The application shall allow the user to pan on the painting within the main viewer. | **M** |
| URF 6.5 | The application shall allow the user to change the contrast of the image displayed in the main viewer. | **M** |
| URF 6.6 | The application shall allow the user to change the saturation of the image displayed in the main viewer. | **M** |
| URF 6.7 | The application shall allow the user to change the brightness of the image displayed in the main viewer. | **S** |
| URF 6.8 | The application shall allow the user to change the gamma correction of the image displayed in the main viewer. | **S** |
| URF 6.9 | The application shall allow the user to change the panning speed within the main viewer. | **S** |
| URF 6.10 | The application shall allow the user to change the zooming speed within the main viewer. | **S** |

Continued on next page

Table 6: Requirements for viewing data (Continued)

| ID | Requirement | Priority |
|---|---|---|
| URF 6.11 | The application shall have the option to reset the current visual settings choices to the default visual settings. | **C** |
| URF 6.12 | The application shall allow the user to view at least two main viewers from one data source at a time. | **C** |
| URF 6.13 | Each main viewer shall have a different corresponding layer stack. | **C** |
| URF 6.14 | When there are two or more main viewers, the application shall allow the user to lock the main viewers together such that panning or zooming in one pans or zooms the other main viewers. | **C** |
| URF 6.15 | When there are two or more main viewers, the application shall allow the user to unlock the main viewers such that panning or zooming in one does not pan or zoom the other main viewers. | **C** |
| URF 6.16 | The application shall restrict the zoom to not be limitless. | **C** |
| URF 6.17 | The application shall allow the user to reset the filters of the image displayed in the main viewer to the original value. | **C** |
| URF 6.18 | The application shall allow users to have multiple data sources open concurrently. | **W** |

### 3.1.6 Exporting data

Table 7: Requirements for exporting data

| ID | Requirement | Priority |
|---|---|---|
| URF 7.1 | The application shall allow the user to export the image that is displayed in the main viewer in a JPEG file format with a resolution of at least 600dpi. | **S** |
| URF 7.2 | The application shall allow the user to export the image/plot that is displayed in any of the visualization views in a JPEG file format with a resolution of at least 600dpi. | **S** |
| URF 7.3 | The application shall allow the user to save the layer configuration. | **C** |

### 3.1.7 General visualization operations

Table 8: Requirements for general visualization operations

| ID | Requirement | Priority |
|---|---|---|
| URF 8.1 | The application shall allow the user to view at least two visualization views on the data at a time. | **M** |
| URF 8.2 | The application shall link the selections and operations made in one view across all active views. | **M** |
| URF 8.3 | The main viewer shall use a constant color-coding system to indicate the links between views. | **M** |

### 3.1.8 Selection tool

Table 9: Requirements for selection tool

| ID | Requirement | Priority |
|---|---|---|
| URF 9.1 | The application shall have a selection tool allowing the user to select a subarea of the painting from the main viewer. | **M** |
| URF 9.2 | The application shall depict a bar chart of the complete elemental composition of the current selection. | **M** |
| URF 9.3 | The application shall depict the average spectrum based on the raw data of the current selection. | **M** |
| URF 9.4 | When using the selection feature within the main viewer, the application shall highlight the current selection. | **M** |
| URF 9.5 | The application shall have a rectangle selection tool. | **M** |
| URF 9.6 | The application shall have a freehand drawing selection tool. | **C** |
| URF 9.7 | The application shall allow the user to change the selection mode. | **C** |
| URF 9.8 | The application shall have the option to clear the current selection. | **C** |

### 3.1.9 Elemental channels

Table 10: Requirements for elemental channels

| ID | Requirement | Priority |
|---|---|---|
| URF 10.1 | The application shall have a elemental channel view for selecting elements to be highlighted. | **M** |
| URF 10.2 | The application shall allow the user to select at least 4 elements to be highlighted simultaneously in the elemental distribution view. | **M** |
| URF 10.3 | The application shall allow the user to select at least 4 elements to be highlighted simultaneously in the element visualization view. | **M** |
| URF 10.4 | The application shall provide a means for the user to select in which color each element is highlighted in the elemental distribution view. | **M** |
| URF 10.5 | The application shall set the color intensity of a highlighted element in a pixel based on the relative concentration of the element in that pixel. | **M** |
| URF 10.6 | When a pixel contains multiple elements which are selected, the elemental distribution view shall color that pixel using a combination of the colors of the elements present in that pixel. | **M** |
| URF 10.7 | The application shall allow the user to deselect elements to be highlighted in the elemental distribution view. | **M** |
| URF 10.8 | The application shall allow the user to set an intensity for each element. | **S** |
| URF 10.9 | For each pixel in the painting and element selected, the application shall set the pixel's brightness based on the user-set intensity for that element and the element's abundance. | **S** |
| URF 10.10 | When the user has selected multiple elements, the application shall allow to set an intensity for each element separately. | **C** |

Continued on next page

Table 10: Requirements for elemental channels (Continued)

| ID | Requirement | Priority |
|---|---|---|
| URF 10.11 | The application shall allow the user to view the elements in alphascale. | **C** |

### 3.1.10 Elemental visualization

Table 11: Requirements for elemental visualization

| ID | Requirement | Priority |
|---|---|---|
| URF 11.1 | The element visualization view shall have the option to view the elemental composition as a bar chart. | **M** |
| URF 11.2 | The elemental bar chart shall visualize the average elemental intensity over the whole painting per element. | **M** |
| URF 11.3 | Within the elemental bar chart, the application shall visualize the elemental composition across the painting of regions selected by the user. | **M** |
| URF 11.4 | The element visualization view shall have the option to view the elemental composition as a line chart. | **S** |
| URF 11.5 | The elemental line chart shall visualize the average elemental intensity over the whole painting per element. | **S** |
| URF 11.6 | Within the elemental line chart, the application shall visualize the elemental composition across the painting of regions selected by the user. | **S** |

### 3.1.11 Lens

Table 12: Requirements for the lens

| ID | Requirement | Priority |
|---|---|---|
| URF 12.1 | The application shall restrict the lens to the main viewer. | **M** |
| URF 12.2 | The application shall allow the user to move the lens within the main viewer. | **M** |
| URF 12.3 | If there are two or more layers, the application shall allow the user to determine which layer the area of the main viewer covered by the lens shall display. | **M** |
| URF 12.4 | The application shall allow the user enable the lens. | **M** |
| URF 12.5 | The application shall allow the user to disable the lens. | **M** |
| URF 12.6 | The application shall allow the user to lock the lens in a specific area while being able to move the cursor to other locations. | **S** |
| URF 12.7 | The application shall allow the user to change the lens size. | **S** |
| URF 12.8 | When there are two or more main viewers and the user hovers over one of the main viewers with the lens, the application shall highlight the corresponding areas in all the others. | **C** |
| URF 12.9 | The application shall allow the user to change the lens opacity. | **C** |

### 3.1.12 Color segmentation view

Table 13: Requirements for the color segmentation view

| ID | Requirement | Priority |
|---|---|---|
| URF 13.1 | The application shall contain a color segmentation view showcasing the color segmentation for each separate element. | **M** |
| URF 13.2 | The application shall contain a color segmentation view showcasing the color segmentation for the whole painting based on the RGB image. | **M** |
| URF 13.3 | The application shall allow the user to select at least one specific color cluster from the color segmentation view. | **M** |
| URF 13.4 | The application shall highlight the areas selected in the color segmentation view in the main viewer. | **M** |
| URF 13.5 | When a color is selected from the color segmentation view, the application shall highlight the high-dimensional data associated with that color in all the other views. | **S** |
| URF 13.6 | The application shall provide a means for the user to combine different color segments into a single one, corresponding to the union of the regions of the combined segments. | **C** |
| URF 13.7 | The application shall allow the user to select at least four specific color clusters simultaneously from the color segmentation view. | **C** |
| URF 13.8 | The application shall allow the user to select the complete painting, and number of colors to compute. | **C** |
| URF 13.9 | The application shall provide the option to select a specific element, a threshold and the number of colors to compute. | **C** |

### 3.1.13 Spectral visualization

Table 14: Requirements for spectral visualization

| ID | Requirement | Priority |
|---|---|---|
| URF 14.1 | The spectral visualization shall visualize the average spectrum of the whole painting. | **M** |
| URF 14.2 | The spectral visualization shall provide a means for the user to visualize the average spectrum of the current selection. | **M** |
| URF 14.3 | The spectral visualization shall show the theoretical emission energies of the elements detected in the spectra. | **S** |
| URF 14.4 | The spectral visualization shall allow the user to choose the element for the theoretical emission visualization. | **S** |
| URF 14.5 | The spectral visualization shall allow the user to zoom in and out in the spectral graph. | **S** |
| URF 14.6 | The spectral visualization shall allow the user to pan around the spectral graph. | **S** |

### 3.1.14 Dimensionality reduction visualization

Table 15: Requirements for computing the dimensionality reduction visualization

| ID | Requirement | Priority |
|---|---|---|
| URF 15.1 | The application shall provide a dimensionality reduction visualization option. | **M** |
| URF 15.2 | The DRS shall utilize the Uniform Manifold Approximation and Projection (UMAP) method with dimensionality reduction on the RGB image. | **M** |
| URF 15.3 | The DRS shall provide the option to perform dimensionality reduction on a subpart of the processed data. | **M** |
| URF 15.4 | The DRS shall allow the user to perform the dimensionality reduction with the selected dataset. | **M** |
| URF 15.5 | The DRS shall provide the option to select a specific element and a threshold. | **S** |
| URF 15.6 | The DRS shall perform the dimensionality reduction to only those pixels for which the selected element has an intensity greater than the selected threshold. | **S** |
| URF 15.7 | When the dimensionality reduction process is executed, the DRS shall indicate its loading state. | **S** |
| URF 15.8 | The DRS shall allow the user to terminate the ongoing dimensionality reduction process. | **S** |
| URF 15.9 | The DRS shall provide the option to perform the dimensionality reduction on a subpart of the raw data. | **C** |
| URF 15.10 | The DRS shall provide the option to change the nearest neighbour and minimum distance parameters and metric used in the UMAP method for dimensionality reduction. | **C** |

Table 16: Requirements for features of the dimensionality reduction visualization

| ID | Requirement | Priority |
|---|---|---|
| URF 16.1 | The DRS shall provide a feature that allows the user to specify the overlay shown on the embedding. | **M** |
| URF 16.2 | The DRS shall support the display of RGB, UV, and X-ray overlays on the embedding. | **M** |
| URF 16.3 | When the dimensionality reduction is finished, the DRS shall show the resulting embedding with the selected overlay on top. | **M** |
| URF 16.4 | The DRS shall have a lasso selection interaction tool. | **M** |
| URF 16.5 | When the user selects an area with the lasso tool in the DRS, the application shall highlight all pixels in the image that are mapped by the dimensionality reduction to a point inside the area selected with the lasso tool. | **M** |
| URF 16.6 | The DRS shall support the display of a density overlay on the embedding. | **S** |
| URF 16.7 | The DRS shall support the display of an elemental intensity overlay on the embedding. | **S** |
| URF 16.8 | The DRS shall have a rectangle selection tool. | **S** |

Continued on next page

Table 16: Requirements for features of the dimensionality reduction visualization (Continued)

| ID | Requirement | Priority |
|---|---|---|
| URF 16.9 | When the user selects an area with the rectangle selection ool in the DRS, the application shall highlight all pixels in the image that are mapped by the dimensionality reduction to a point inside the area selected with the rectangle selection tool. | **S** |
| URF 16.10 | When the lens is active, the DRS shall highlight all points in the embedding that correspond to a pixel inside the lens. | **C** |
| URF 16.11 | When an area with the selection tool is selected, the DRS shall highlight all points in the embedding that correspond to a pixel inside the area selected by the selection tool. | **C** |
| URF 16.12 | The application shall provide a means for the user to select in which color the DRS selection is highlighted main viewer. | **C** |

### 3.1.15 Context visualization

Table 17: Requirements for context visualization

| ID | Requirement | Priority |
|---|---|---|
| URF 17.1 | The application shall have a context visualization view. | **S** |
| URF 17.2 | The context visualization view shall display the zoomed-in RGB image at the position of the lens. | **S** |
| URF 17.3 | The application shall allow the user to zoom in and out in the context visualization. | **S** |

### 3.1.16 Themes

Table 18: Requirements for application theming

| ID | Requirement | Priority |
|---|---|---|
| URF 18.1 | The application shall allow the user to choose between light and dark mode. | **C** |
| URF 18.2 | The application shall, by default, use the theme that the user's operating system uses. | **C** |

### 3.1.17 Documentation

Table 19: Requirements for the documentation

| ID | Requirement | Priority |
|---|---|---|
| URF 19.1 | The application shall provide a webpage dedicated to documentation. | **M** |
| URF 19.2 | The documentation shall include an overview of permissible file types that may be used in the application. | **M** |

Continued on next page

Table 19: Requirements for the documentation (Continued)

| ID | Requirement | Priority |
|---|---|---|
| URF 19.3 | The documentation shall provide instructions on how to navigate the application. | **M** |
| URF 19.4 | The documentation shall provide explanations of the graphs used in the application. | **M** |
| URF 19.5 | The documentation shall provide an explanation of the application's API. | **M** |
| URF 19.6 | The documentation shall include an overview of the application's front-end code. | **M** |
| URF 19.7 | The documentation shall provide an overview of the application's back-end code. | **M** |
| URF 19.8 | The documentation shall provide an overview of all keyboard shortcuts available in the application. | **C** |

## 3.2 Constraint requirements

In this section the constraint requirements are specified. Each requirement has a unique ID, starting with URC (User Requirements Constrain), indicating that these are the constraints of the application.

### 3.2.1 Licensing

Table 20: Requirements for licensing

| ID | Requirement | Priority |
|---|---|---|
| URC 1.1 | The application shall be made available under an open source license. | **M** |
| URC 1.2 | The application shall be made available under the GPLv3 or MIT license. | **C** |
| URC 1.3 | The application shall solely depend on dependencies that are licensed under an open source license. | **C** |

### 3.2.2 Usability and version

Table 21: Requirements for usability and version

| ID | Requirement | Priority |
|---|---|---|
| URC 2.1 | The application client shall run on Chrome version 124 or higher. | **M** |
| URC 2.2 | The application client shall run on Firefox version 125 or higher. | **M** |
| URC 2.3 | The application shall run on x86-64 based architectures within a Python environment on the Windows 10, Windows 11 and Ubuntu Server LTS 24.04 operating systems. | **M** |
| URC 2.4 | The application's UI shall use the English language. | **M** |
| URC 2.5 | The application shall run on computer screens. | **M** |
| URC 2.6 | The application shall run on x86-64 based architectures within a Python environment on the macOS operating system. | **S** |
| URC 2.7 | The application client shall run on Safari version 17 or higher. | **S** |
| URC 2.8 | The application shall run on tablet devices. | **C** |
| URC 2.9 | The application shall run within a Docker container. | **C** |

### 3.2.3 Performance and reliability

Table 22: Requirements for performance and reliability

| ID | Requirement | Priority |
|---|---|---|
| URC 3.1 | When a layer is enabled in the layer view and all data for that layer is at the client's side, the main viewer shall update the interactive visualisation within 1 seconds. | **S** |
| URC 3.2 | When a layer is disabled in the layer view, the main viewer shall update the interactive visualisation within 1 seconds. | **S** |

Continued on next page

Table 22: Requirements for performance and reliability (Continued)

| ID | Requirement | Priority |
|---|---|---|
| URC 3.3 | When a color segmentation, context visualization, element visualization, selection bar graph, or Spectral visualization view is made, the application shall display the created view within 1 second. | **S** |
| URC 3.4 | The application shall synchronize selections across active views within 1 second of having made the selection. | **S** |

# A USE CASES

## A.1 Data source and exporting related use cases

### A.1.1 Uploading a data source

| | |
|---|---|
| **Summary** | The actor selects to upload a data source and fills in all the upload fields. The data source is added to the server and becomes visible in the data source list |
| **Preconditions** | The actor is on any screen of the application. |
| **Postconditions** | The data source gets added to the server and is visible to the actor in the data source list. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | To provide the actor with an environment in which they can upload their painting data for further analysis. |
| **Related requirements** | URF 3.1, 3.2, 3.3, 3.5, 3.7, 3.8 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor navigates to the upload section. | |
| 2 | | The system provides the actor with the necessary fields to create the data source. |
| 3 | The actor enters a name for the data source, uploads the data source files, and enters names for the individual components in the data source. | |
| 4 | | The system adds the data source to the server and adds it to the list of available data sources. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 3A | The actor selects files which do not conform to the file type requirement and uploads them. | |
| 4A | | The system notifies the actor about the file type chosen not being correct and having to choose their files again. Return to step 3. |

| *ALTERNATIVE FLOW* | |
|---|---|
| 4B | If the actor has left any of the mandatory fields empty, the application shall notify the actor. Return to step 3. |

| *ALTERNATIVE FLOW* | | |
|---|---|---|
| 3C | The actor enters a name for the data source that is already in use by another data source. | |
| 4C | | The system notifies the actor that the chosen name is already in use and will have to choose a different name. Return to step 3. |

| *ALTERNATIVE FLOW* | | |
|---|---|---|
| 3D | The actor enters a name for a data source component that is already in use by another data source component. | |
| 4D | | The system notifies the actor that the chosen name is already in use and will have to choose a different name. |

### A.1.2 Editing a data source

| | |
|---|---|
| **Summary** | The actor selects a data source to edit and edits its configuration, the changed configuration gets saved in the server and is visible in the data source list |
| **Preconditions** | There is at least one data source. The actor is viewing the data source list. |
| **Postconditions** | The actor-specified changes to the data source are saved in the server and are visible to the actor in the data source list. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Could have |
| **Goal** | Giving the actor the flexibility to adapt already existing data sources by being able to modify the name and contents of the configuration. This is to enhance the research process in case new scans should be added. |
| **Related requirements** | URF 3.2, 3.10, 3.11, 3.12 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor requests to edit a data source. | |
| 2 | | The system opens the data configuration window for the desired data source. |
| 3 | The actor makes the desired changes and saves the configuration. | |
| 4 | | The system saves the changes the actor made to the data source. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 3A | The actor selects files which do not conform to the file type requirement. | |
| 4A | | The system notifies the actor that the file type chosen is not correct. Return to step 3. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 3B | The actor chooses a name for the data source that is already in use by another data source. | |
| 4B | | The system notifies the actor that the chosen name is already in use. Return to step 3. |

### A.1.3 Deleting a data source

| | |
|---|---|
| **Summary** | The actor requests to delete a data source and the data source is removed from the server and the data source list. |
| **Preconditions** | There is at least one data source. The actor is viewing the data source list. |
| **Postconditions** | The data source is removed from the server and the data source list. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Should have |
| **Goal** | Giving the actor the ability to remove data sources that they no longer require, allowing for a more organised working environment. |
| **Related requirements** | URF 3.9 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor requests to delete a data source. | |
| 2 | | The system shows a message asking whether the actor is sure they want to remove the data source. |
| 3 | The actor confirms they want to remove the data source. | |
| 4 | | The system removes the data source from the server and the data source list. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 3A | The actor cancels their action to remove the data source. | |
| 4A | | The system does not remove the data source from the server and from the list of available data sources. |

### A.1.4 Exporting the viewer's image

| | |
|---|---|
| **Summary** | The actor clicks the button to export the image currently in the viewer, after which the image gets exported to the actor's device. |
| **Preconditions** | The actor has loaded in the data source of the painting they want to export. |
| **Postconditions** | The image is available on the actor's device. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Should have |
| **Goal** | Allowing the actor to export their generated images such that they can be used as visual aids in presentations, posters, etc. |
| **Related requirements** | URF 7.1 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor clicks the button to export the image currently visible in the viewer. | |
| 2 | | The system exports the image in the viewer to the actor's device. |

### A.1.5 Exporting the image from one of the visualization views

| | |
|---|---|
| **Summary** | The actor clicks the button to export the image currently displayed in a visualization view, the image gets exported to the actor's device. |
| **Preconditions** | The actor has loaded a data source and is viewing the visualization view for which they want to export the visualization. |
| **Postconditions** | The image is available on the actor's device. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Should have |
| **Goal** | Allowing the actor to export their generated charts such that they can be used as visual aids in presentations, posters, etc. |
| **Related requirements** | URF 7.2 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor clicks the button to export the desired visualization in the visualization view. | |
| 2 | | The system exports the desired visualization to the actor's device. |

## A.2 General functionality use cases

### A.2.1 Changing the order of layers.

| | |
|---|---|
| **Summary** | The actor drags a layer to a new position and the layer stack and main viewer are updated in accordance to this change in layer position. |
| **Preconditions** | The actor is in the layers view, and the main viewer related with that layer view is open. There are at least two layers present. |
| **Postconditions** | The layer stack is updated in accordance to the new position of the dragged layer. The main viewer is updated to reflect the change in the layer stack. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | Providing an adaptable and intuitive environment for changing what's visible in the main viewer. |
| **Related requirements** | URF 5.1, 5.2 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor drags the image layer to the position they desire. | |
| 2 | | The system reorders the layers in the layer stack according to the position of the dragged layer. |
| 3 | | The main viewer is updated to reflect the changes in the layer stack. |

### A.2.2 Changing the opacity of a layer.

| | |
|---|---|
| **Summary** | The actor changes the opacity of a layer and the opacity value gets applied to the layer. The main viewer is updated to reflect the changes in the layer's opacity. |
| **Preconditions** | The actor is in the layers view, and the main viewer related with that layer view is open. There is at least one layer. |
| **Postconditions** | The opacity of the layer is updated to the actor-specified opacity. The main viewer is updated to reflect the change in the layer's opacity. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | Providing an adaptable and intuitive environment for changing what's visible in the viewer. |
| **Related requirements** | URF 5.3, 5.4 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor changes the opacity to the desired valued for the desired layer. | |
| 2 | | The system applied the new opacity value to the layer. |
| 3 | | The main viewer gets updated to reflect the changes in the layer's opacity. |

### A.2.3 Resetting the opacity of a layer.

| | |
|---|---|
| **Summary** | The actor resets the opacity of a layer and the opacity value gets applied to the layer. The main viewer is updated to reflect the changes in the layer's opacity. |
| **Preconditions** | The actor is in the layers view, and the main viewer related with that layer view is open. There is at least one layer. |
| **Postconditions** | The opacity of the layer is reset to the default opacity value. The main viewer is updated to reflect the change in the layer's opacity. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Could have |
| **Goal** | To allow the actor to easily revert the opacity value of a layer to improve the exploration between different layers. |
| **Related requirements** | URF 5.10 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor request to reset the opacity for the desired layer. | |
| 2 | | The system resets applied the default opacity value to the layer. |
| 3 | | The main viewer gets updated to reflect the changes in the layer's opacity. |

### A.2.4 Enabling/Disabling a layer in the layer stack

| | |
|---|---|
| **Summary** | The actor enables/disables the layer in the layer stack, which they want to show/hide in the main viewer. |
| **Preconditions** | The actor is in the layers view, and the main viewer related with that layer view is open, and there is at least one layer in the layer stack. |
| **Postconditions** | The layer is enabled/disabled and the main viewer is updated in accordance. |
| **Actors** | Any User |
| **Trigger** | - |
| **Priority** | Must Have |
| **Goal** | Providing an adaptable and intuitive environment for changing what's visible in the main viewer. |
| **Related requirements** | URF 5.4, 5.5, 5.6 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor requests to enable/disable a layer in the layer stack. | |
| 2 | | The system toggles the visibility of the layer to the actor's desired visibility and updates the main viewer. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 1A | The actor request to enable a layer that is enabled. | |
| 2A | | The system disables the layer and updates the main viewer. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 1A | The actor request to disable a layer that is disabled. | |
| 2A | | The system enables the layer and updates the main viewer. |

### A.2.5 Enabling two main viewers.

| | |
|---|---|
| **Summary** | The actor requests another main viewer besides the currently active main viewer and the system updates the application to show two main viewers. |
| **Preconditions** | The actor has one main viewer open and is viewing this. |
| **Postconditions** | The system shows two main viewers at the same time to the actor. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Could have |
| **Goal** | To allow the actor to compare different sections of the same painting at the same time. |
| **Related requirements** | URF 6.12 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor request a second main viewer. | |
| 2 | | The system shows a second main viewer to the actor. |

### A.2.6 Using the lens tool

| | |
|---|---|
| **Summary** | The actor selects the lens viewing mode, and the lens viewing mode gets enabled around the actor's cursor; the actor chooses the layer that the lens should display, and the lens displays this layer. |
| **Preconditions** | The actor has loaded in the data source of the painting they want to explore, and there are at least two layers active. |
| **Postconditions** | The area covered by the lens displays the layer the actor has chosen. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | Providing an easy-to-use method for directly comparing two image layers. |
| **Related requirements** | URF 12.3, 12.4 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects the lens viewing mode and the layer to view through the lens. | |
| 2 | | The system enables the lens viewing mode. Around the actor's cursor, there is now a lens. |
| 3 | | The area covered by the lens displays the layer the actor has chosen. |

### A.2.7 Changing the lens' size

| | |
|---|---|
| **Summary** | The actor changes the size of the lens. The lens viewing mode is updated to reflect the changes in the lens size. |
| **Preconditions** | The actor is in the lens viewing mode, and there are at least two layers active. |
| **Postconditions** | The area visible through the lens changes in size. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Should have |
| **Goal** | Allowing the actor to change the size of the lens. |
| **Related requirements** | URF 12.7 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor changes the lens size. | |
| 2 | | The system applies the new size value to the lens. |
| 3 | | The lens viewing mode gets updated to reflect the changes in the lens size. |

### A.2.8 Changing the lens' opacity

| | |
|---|---|
| **Summary** | The actor changes the opacity of the lens. The lens viewing mode is updated to reflect the changes in the opacity. |
| **Preconditions** | The actor is in the lens viewing mode, and there are at least two layers. |
| **Postconditions** | The layer visible through the lens changes in opacity. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Could have |
| **Goal** | Allowing the actor to change the opacity of the lens. |
| **Related requirements** | URF 12.9 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor changes the lens' opacity. | |
| 2 | | The new opacity value is applied to the lens. |
| 3 | | The lens viewing mode gets updated to reflect the changes in the lens' opacity. |

### A.2.9 Using a selection tool

| | |
|---|---|
| **Summary** | The actor chooses the desired selection tool and selects with this tool the desired area in the main viewer. The pixels corresponding to the current selection are highlighted in the main viewer. |
| **Preconditions** | The actor has loaded in the data source of the painting they want to explore. |
| **Postconditions** | The selected areas are highlighted in the main viewer. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must/could have |
| **Goal** | To provide the actor with the ability to select an area that they are interested in being visualized in the available charts. |
| **Related requirements** | URF 9.5, 9.6 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects the desired selection tool. | |
| 2 | The actor selects the desired area in the main viewer using the selected selection tool. | |
| 3 | | The system highlights the corresponding pixels in the main viewer. |
| 4 | | The chart visualization views are updated in accordance with the selected areas. |

### A.2.10 Changing the filter on the image.

| | |
|---|---|
| **Summary** | The actor changes the filter of the image and the main viewer is updated to reflect the changes in the filter. |
| **Preconditions** | The actor has loaded in the data source of the painting they want to explore. |
| **Postconditions** | The filter on the image is updated to the actor-specified filter options. The main viewer is updated to reflect the changes in the filter. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | To offer the actor a customizable viewing experience. |
| **Related requirements** | URF 6.5, 6.6, 6.7, 6.8 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor changes the filter settings of the image. | |
| 2 | | The system changes the filter of the image to the actor-specified settings. |
| 3 | | The main viewer is updated according to the new filter. |

### A.2.11 Resetting the filters of the image.

| | |
|---|---|
| **Summary** | The actor resets the filters of the image and the main viewer is updated to reflect the changes in filters' values. |
| **Preconditions** | The actor has loaded in the data source of the painting they want to explore. |
| **Postconditions** | The filters of the image are updated to the original values. The main viewer is updated to reflect the change in filter values. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Could have |
| **Goal** | To allow the actor to easily revert all visual changes made to the painting. |
| **Related requirements** | URF 6.17 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor specifies the filter that should be reset to their default values. | |
| 2 | | The system changes the specified filter's value of the image to the default value. |
| 3 | | The main viewer is updated according to the new, default, values. |

## A.3 Exploration-related use cases

### A.3.1 Generating color segments using color segmentation

| | |
|---|---|
| **Summary** | The actor selects on which they want to perform color segmentation and chooses the parameters used. The system then generates the color segments using the actor's desired settings and shows the segments in the color segmentation view. |
| **Preconditions** | There is an open color segmentation view |
| **Postconditions** | The color segments from the color segmentation with the actor's specified settings is shown in the color segmentation view. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | To enable the actor to explore colors present in the painting based on their desired settings. |
| **Related requirements** | URF 13.1, 13.2, 13.8, 13.9 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects on which they want to perform color segmentation and chooses the parameters used in the color segmentation method. | |
| 2 | The actor request to perform the color segmentation with the specified settings. | |
| 3 | | The system generates the color segmentation with the actor's desired settings. |
| 4 | | The system shows the color segments in the color segmentation view. |

### A.3.2 Exploring color segments from color segmentation

| | |
|---|---|
| **Summary** | The actor selects color segments in the color segmentation view, and the system highlights the pixels in the main viewer corresponding to the selected color segments. |
| **Preconditions** | The actor is in the colors segmentation view with at least one color segment present. |
| **Postconditions** | The pixels corresponding to the selected color segments are highlighted in the main viewer. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | To enable the actor to clearly see the (location) of the pixels related to a color segment while maintaining visibility of the surrounding context. |
| **Related requirements** | URF 13.3, 13.4, 13.7 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects the desired color segments in the color segmentation view. | |
| 2 | | The system highlights the corresponding pixels to the color segments that the actor selected in the main viewer. |

### A.3.3 Viewing the presence of elements

| | |
|---|---|
| **Summary** | The actor selects the elements they want to explore the presence of in the elemental distribution view. The system highlights the areas in the main viewer with a sufficient presence of the selected elements. |
| **Preconditions** | The actor is in the elemental channels view and there is at least one elemental channel present. |
| **Postconditions** | The areas in which the selected elements are sufficiently present are highlighted in the main viewer. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | Providing a method for the actor to allow them to see the presence of and correlation between elements within a painting. |
| **Related requirements** | URF 10.1, 10.2, 10.6 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects the elements of which they want to explore the presence in the painting. | |
| 2 | | The system highlights the pixels in the main viewer in which the selected elements in elemental distribution view are sufficiently present. |

### A.3.4 Deselecting elements

| | |
|---|---|
| **Summary** | The actor deselects the elements they want to stop exploring the presence of in the elemental distribution view. The system stops highlighting the areas in which the deselected elements are sufficiently present. |
| **Preconditions** | The actor is in the elemental channel view. At least one element is selected and is being highlighted in the elemental distribution view. |
| **Postconditions** | The areas in which the deselected elements are present, are not being highlighted in the elemental distribution view. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | Creating a clear image in the main viewer in which no unnecessary elements and areas are highlighted. |
| **Related requirements** | URF 10.7 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor deselects the elements they want to stop exploring the presence of in the painting from the elemental channel view. | |
| 2 | | The system stops highlighting the areas of the painting in which the deselected elements in the elemental distribution view are present. |

### A.3.5 Changing the color for an element

| | |
|---|---|
| **Summary** | The actor changes the color of an element, in the elemental distribution view, the color of the specific element changes to the color decided by the actor. |
| **Preconditions** | The actor is in the elemental channel view and at least one element is selected. |
| **Postconditions** | The areas in which the selected element is presented are highlighted in the color specified by the actor in the elemental distribution view. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | Improving the exploration of the elemental composition of a painting. |
| **Related requirements** | URF 10.4 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects in which color they want the selected element to be highlighted in. | |
| 2 | | The system highlights the areas in the painting in which the selected element is present in the actor's selected color. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 2A | | Other selected elements are present in the highlighted area. |
| 3A | | The displayed color is a combination of all selected colors present in the area. |

### A.3.6 Changing the Intensity of an element

| | |
|---|---|
| **Summary** | The actor changes the intensity of an element, in the elemental distribution map the areas highlighted comply with the selected element and its intensity. |
| **Preconditions** | The actor is in the elemental channel view and there is at least one selected element. |
| **Postconditions** | The areas in which the selected element is present, are highlighted in the elemental distribution view in compliance with the actor-specified intensity. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Should have |
| **Goal** | To allow the actor to better visualize the presence of elements. |
| **Related requirements** | URF 10.8 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor changes the intensity of the selected element. | |
| 2 | | The highlighted areas in the elemental distribution view which are respective to the selected element, change in visibility in accordance with the actor-specified intensity. |

### A.3.7 Viewing the elemental composition in the element visualization view

| | |
|---|---|
| **Summary** | The actor requests to see the elemental composition in the painting and the system shows this in a graph in the element visualization view. |
| **Preconditions** | The actor is in the element visualization view. |
| **Postconditions** | The actor sees the elemental composition. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | To allow the actor to see the elemental composition in the whole or in a part of the painting. |
| **Related requirements** | URF 11.1, 11.2, 11.3, 11.4, 11.5, 11.6 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor requests to see the elemental composition within the whole painting or within the current selection. | |
| 2 | | The system updates the element visualization view to show the elemental composition within the actor's requested area. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 1A | The actor requests to see the elemental composition within the current selection, but there is no current selection in the main viewer. | |
| 2A | | The system updates the element visualization view to show the elemental composition within the whole painting. |

### A.3.8 Viewing the average spectrum in the spectral visualization

| | |
|---|---|
| **Summary** | The actor requests to see the average spectrum in the whole painting or in the current selection and the system updates the spectral visualization to show this. |
| **Preconditions** | The actor is in the spectral visualization. |
| **Postconditions** | The graph in the spectral visualization is updated to show the average and spectrum in the whole painting or in the current selection. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | To allow the actor to explore the average spectrum of the whole or of a part of the painting. |
| **Related requirements** | URF 14.1, 14.2 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor requests to see the average spectrum of the whole painting or the current selection. | |
| 2 | | The system updates the spectral visualization to show the average spectrum of the actor's requested area. |

### A.3.9 Viewing the theoretical spectrum in the spectral visualization

| | |
|---|---|
| **Summary** | The actor requests to see the theoretical spectrum of an element and the system updates the spectral visualization to show this. |
| **Preconditions** | The actor is in the spectral visualization. |
| **Postconditions** | The graph in the spectral visualization is updated to show the theoretical spectrum of the element. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Should have |
| **Goal** | To allow the actor to compare the average spectrum with the theoretical spectrum. |
| **Related requirements** | URF 14.3, 14.4 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects an element for which they want to see the theoretical spectrum. | |
| 2 | The actor requests to see the theoretical spectrum of the selected element. | |
| 3 | | The system updates the spectral visualization to show the theoretical spectrum of the actor's selected element. |

### A.3.10 Exploring using dimensionality reduction

| | |
|---|---|
| **Summary** | The actor chooses the element, threshold and overlay they want to explore and this is reflected in the dimensionality reduction visualization. |
| **Preconditions** | The actor is in the dimensionality reduction view. |
| **Postconditions** | The dimensionality reduction visualization is updated according to the actor's choices of settings. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | To provide the actor with the ability to explore regions that are similar in elemental distribution. |
| **Related requirements** | URF 15.1, 15.3, 15.5, 15.6, 15.4, 16.1, 16.2, 16.6, 16.7, 16.3 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects a combination of element (or complete painting), threshold and overlay they want to explore. | |
| 2 | | The system computes the embedding using dimensionality reduction with the given settings. |
| 3 | | The system shows the visualization of the generated embedding with the actor's selected overlay. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 2A | | There is already an embedding satisfying the actor's desired settings. Go to step 3. |

| | | |
|---|---|---|
| *ALTERNATIVE FLOW* | | |
| 2B | The actor aborts the dimensionality reduction. | |
| 3B | | The system shall display an error stating the computation of the dimensionality reduction visualization failed. |

### A.3.11 Selection interaction within the dimensionality reduction

| | |
|---|---|
| **Summary** | The actor selects the desired selection tool and selects the desired area in the dimensionality reduction visualization. The corresponding pixels are highlighted in the main viewer. |
| **Preconditions** | The actor is in the dimensionality reduction view and a dimensionality reduction visualization is being shown. |
| **Postconditions** | The corresponding pixels to the selected areas are highlighted in the main viewer. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Must have |
| **Goal** | To provide the actor with the ability to select a flexible area that they are interested in exploring within the dimensionality reduction visualization. |
| **Related requirements** | URF 16.4, 16.5, 16.8, 16.9 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor chooses a selection tool. | |
| 2 | The actor selects the desired area in the dimensionality reduction visualization using the selection tool. | |
| 3 | | The system highlights the pixels corresponding to the pixels in the selected area in the main viewer. |

### A.3.12 Changing the highlighting color for dimensionality reduction

| | |
|---|---|
| **Summary** | The actor chooses a highlight color for the selected area in the dimensionality reduction window, the main viewer is updated to show the highlighted areas in the actor specified color. |
| **Preconditions** | The actor is in the dimensionality reduction view and has made a selection in the dimensionality reduction window using the lasso tool or rectangle selection tool. |
| **Postconditions** | The selected areas are highlighted in the main viewer in the color specified by the actor. |
| **Actors** | Any user |
| **Trigger** | - |
| **Priority** | Could have |
| **Goal** | To provide the actor with the ability to change the highlighting color to better explore the presence of the highlighted area. |
| **Related requirements** | URF 16.12 |

| Step | Actor actions | System actions |
|---|---|---|
| *MAIN FLOW* | | |
| 1 | The actor selects in which color they want the selected area in the dimensionality reduction window to be highlighted in. | |
| 2 | | The areas in the main viewer corresponding to the selected area in the dimensionality reduction window, are highlighted in the actor specified color. |

# B SIGNING PAGE

Hereby the client, the supervisor and the development team agree upon this document.

| **Client** | **Supervisor** |
| --- | --- |
| *Name* | *Name* |
| | |
| *Date* | *Date* |
| | |
| *Signature* | *Signature* |
| | |