JPEG Linked Media Format (JLINK) Applications

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Víctor Ojeda Franco

In partial fulfilment

of the requirements for the degree in

TELECOMMUNICATIONS TECHNOLOGIES AND SERVICES ENGINEERING

Advisor: Silvia Llorente Viejo

Barcelona, June 2022

Abstract

The growing and emerging of the large use of images in recent decades has fostered the need to innovate with new functionalities and applications. That is the reason why the Joint Photographic Experts Group (JPEG) is currently developing new standards.

This thesis focuses on the JPEG Systems Part 7: JPEG Linked Media Format (JLINK) standard, which aims to allow a set of related images to be encapsulated in a single, and, through an adapted viewer, to move between images through interactive points.

The main objective of the project is to implement an application that allows to create, visualize and modify this type of files defined by the specifications of the standard.

The result has been a web application capable of performing the aforementioned actions, with the addition of more functions. It also has a database to have control of the stored files.

The usefulness of defining a standard has been proven, and the importance of developers to check that the specifications given are correct or need to be modified.

Resum

El creixement i l'aparició de l'ampli ús de les imatges a les darreres dècades ha fomentat la necessitat d'innovar amb noves funcionalitats i aplicacions. Per això, el Joint Photographic Experts Group (JPEG) està desenvolupant actualment nous estàndards.

Aquesta tesi se centra en l'estàndard JPEG Systems Part 7: JPEG Linked Media Format (JLINK), que pretén permetre encapsular un conjunt d'imatges relacionades en una de sola, i, mitjançant un visor adaptat, moure's entre les imatges mitjançant punts interactius.

L'objectiu principal del projecte és implementar una aplicació que permeti crear, visualitzar i modificar aquest tipus de fitxers definits per les especificacions de l'estàndard.

El resultat ha estat una aplicació web capaç de fer les accions esmentades, amb l'afegit de més funcions. També disposa d'una base de dades per tenir el control dels fitxers emmagatzemats.

S'ha comprovat la utilitat de la definició d'un estàndard i la importància que els desenvolupadors comprovin que les especificacions donades són correctes o s'han de modificar.

Resumen

El creciente y emergente uso de imágenes en las últimas décadas ha fomentado la necesidad de innovar con nuevas funcionalidades y aplicaciones. Por ello, el Joint Photographic Experts Group (JPEG) está desarrollando actualmente nuevos estándares.

Esta tesis se centra en el estándar JPEG Systems Part 7: JPEG Linked Media Format (JLINK), que pretende permitir encapsular un conjunto de imágenes relacionadas en una sola, y, a través de un visor adaptado, moverse entre las imágenes mediante puntos interactivos.

El objetivo principal del proyecto es implementar una aplicación que permita crear, visualizar y modificar este tipo de archivos definidos por las especificaciones del estándar.

El resultado ha sido una aplicación web capaz de realizar las acciones mencionadas, con el añadido de más funciones. También dispone de una base de datos para tener el control de los ficheros almacenados.

Se ha comprobado la utilidad de la definición de un estándar y la importancia de que los desarrolladores comprueben que las especificaciones dadas son correctas o deben ser modificadas.

Acknowledgements

I would like to thank my supervisor, Silvia Llorente, for allowing me to work on this project, guiding me and offering her help whenever I needed it.

Revision history and approval record

Revision	Date	Purpose
0	09/05/2022	Document creation
1	20/06/2022	Document revision
2	21/06/2022	Document approval

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Víctor Ojeda Franco	victor.ojeda.franco@gmail.com victor.ojeda@estudiantat.upc.edu
Silvia Llorente Viejo	silvia.llorente@upc.edu

Written by:		Reviewed and approved by:	
Date	09/05/2022	Date	21/06/2022
Name	Víctor Ojeda Franco	Name	Silvia Llorente Viejo
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables:

List of Formulas:

1. Introduction

The use of images has increased in the last decade, thanks to the great growth of social networks, boosting photography, digital art and more multimedia content. That is why it has been necessary to add more functionality and information to the content of the images.

Based on this premise the Joint Photographic Experts Group (JPEG) [1], the main committee of specialists that devised one of the most widely used digital image encoding and compression standards since its creation, has worked on a systematic review and consolidation of its file formats, functionalities and code stream syntax. The goal was and is to define an overall framework for future and legacy standards called JPEG Systems [2] to ensure interoperability and functionality exchange between these standards.

Like most JPEG standards, it is divided into different parts, each with a specific function, such as data encapsulation, transport, High Dynamic Range (HDR) tools, privacy and security, definition of image metadata, 360 degree images, augmented reality and 3D, among others.

This project is focused on JPEG Systems Part 7: JPEG Linked Media Format (JLINK) [3], which is currently under development. The main idea that defines the standard is to allow encapsulating in the same file a set of images that have a relationship between them. Moreover, it defines the transition between images by means of embedded access points that can be followed to the using a viewer capable of reading this type of files.

To design and implement the structure of this new file, another part of JPEG Systems, Part 5: JPEG Universal Metadata Box Format (JUMBF) [4] is used. It specifies a framework for JPEG standards to add universal metadata embedded in the file, for future applications to provide more features.

The Distributed Multimedia Applications Group (DMAG) [5] of the Computer Architecture Department of the UPC is contributing to the specification of these new standards.

This thesis consists of the analysis of the previously mentioned standards, and the implementation of a web application capable of performing several actions on JLINK files. For this purpose, the Reference Software [6] already available from DMAG has been updated and used. During the project, problems and improvements in JLINK have been discovered and contributed to the JPEG standardization committee.

To this end, an analytical reading of the specifications defined by the standards in question has been carried out, and upon completion of the project, the importance of the standardization process and the organizations that are part of it has been comprehended.

The project also takes into account how to carry out the process of implementing a web development, the different tasks and roles needed to reach the expected result. While developing, the author has used different technologies and programming languages to ensure that both the functionalities and the user experience are correct.

This document presents the planning and execution of the project realization process, an analysis of the technology used in the implementation and a description of the developed web application. It is followed by the results obtained, the cost of the project, the environmental and social impact and finally the conclusions and future development ideas that can follow this thesis.

1.1. Statement of purpose

The main objective of this project is to implement a web application that works with JPEG Linked Media Format (JLINK) [3] and JPEG Universal Metadata Box Format (JUMBF) [4] standards. The new application has to be capable of creating, viewing and modifying the structure that defines the standard. The application also has to interact with a database to save, extract and delete data associated to the different images in the system.

1.2. Requirements and specifications

Project requirements:

- The result of the project should accomplish the goals formerly stated in order to generate applications that use JLINK.
- The application has to be able to show a visualization of a JLINK file.
- The application should also create and modify JLINK files.
- The application should also interact with a relational database.

Project specifications:

- The client side of the web application is to be coded with HTML, CSS and Javascript.
- The server side of the web application is to be coded in Java language, since it will use Servlets, and Java classes are going to be used to implement several utilities.

1.3. Methods and procedures

This project presented here is intended as a continuation of the work done in the thesis "JPEG Universal Metadata Box Format and JPEG Linked Media Format: a Reference Software Implementation" [6] carried out by Carlos Valverde Diaz and supervised by Jaime Delgado Mercé in 2021. The code implemented in the former project will be used as the base tool for the development of this degree final thesis (TFG).

The project consists of a web application coded using NetBeans IDE. The implementation of the actions that the application must accomplish lays on the ISO/IEC 19566-7 (JLINK) [3], and in order to understand how JLINK works it is also necessary to take into account the document ISO/IEC 19566-5:2018 (JUMBF) [4].

The procedure for achieving the project objectives has had the following steps:

- Definition of an initial approach and outline of the web application.
- Development of the web application designed.
- Test of the application to check everything works well or find bugs to fix.
- Demonstrations during meetings with the project supervisor, Silvia Llorente, of the implemented advances.

For the tracking of the meetings with the project supervisor and the time planning, the workspace offered by the Notion [7] application has been used.

1.4. Work plan

The planning carried out at the beginning of the project has not differed much from the planning performed during the project.

In the first work package, the software infrastructure on which the project implementation has been developed was also assembled. In task T1.1 the "Reference Software for both JUMBF and JLINK standards" project result was launched, and the code flow was studied in order to understand it. To perform task T1.2 a test web application was made and after testing it was verified that the application to be developed could be via web.

The tasks dedicated to WP3 have been changed to those that were done. The main difference that we found is that in the initial proposal the blocks dedicated to development were performed almost sequentially, during the implementation we realized that WP3 and WP4 should be performed in parallel, to work at the same time both in client and server side in the JLINK viewer.

During the implementation of WP5, improvements were made to what was implemented in WP2.

A test task was performed for the three actions that the project performs on the JLINK standard at the end of each of them, and another test task on the finished web application.

The work blocks resulting from the project process are as follows:

Table 1. Project Work Packages

<table>
<tr><td colspan="2">Project:
Use the "Reference Software for both JUMBF and JLINK standards".</td><td>WP ref: 1</td></tr>
<tr><td colspan="2">Major constituent: Reference Software for both JUMBF and JLINK standards.</td><td>Sheet 1 of 4</td></tr>
<tr><td colspan="2" rowspan="2">Short description:
Study and use the code of "Reference Software for both JUMBF and JLINK standards” and verify that it can be implemented in a web application.</td><td>Planned start date: 28/02/2022
Planned end date: 13/03/2022</td></tr>
<tr><td>Start event: 28/02/2022
End event: 15/03/2022</td></tr>
<tr><td>Internal task T1.1: Run the “Reference Software for both JUMBF and JLINK standards”.
Internal task T1.2: Verify that the application can be implemented as a web application.</td><td>Deliverables:</td><td>Dates:</td></tr>
</table>

<table>
<tr><td colspan="2">Project:
Initial design and implement JLINK file creator.</td><td>WP ref: 2</td></tr>
<tr><td colspan="2">Major constituent: JLINK file creator.</td><td>Sheet 2 of 4</td></tr>
<tr><td colspan="2" rowspan="2">Short description:
Make an initial design of the application to be developed and implement a JLINK file creator, based on the code of “Reference Software for both JUMBF and JLINK standards”.</td><td>Planned start date: 14/03/2022
Planned end date: 27/03/2022</td></tr>
<tr><td>Start event: 14/03/2022
End event: 27/03/2022</td></tr>
<tr><td>Internal task T2.1: Initial design.

Internal task T2.2: Implement JLINK file creator.</td><td>Deliverables:
Proposal System Block Diagram

JLINK file creator</td><td>Dates:
20/03/2022

27/03/2022</td></tr>
</table>

<table>
<tr><td colspan="2">Project:
Implement back-end JLINK file viewer.</td><td>WP ref: 3</td></tr>
<tr><td colspan="2">Major constituent:
JLINK viewer.</td><td>Sheet 2 of 4</td></tr>
<tr><td colspan="2" rowspan="2">Short description:
Implement JLINK file viewer server development, capable of displaying the root image and the transition to the branch images.</td><td>Planned start date: 28/03/2022
Planned end date: 24/04/2022</td></tr>
<tr><td>Start event: 12/03/2022
End event: 7/03/2022</td></tr>
<tr><td>Internal task T3.1: Parse the JLINK File.

Internal task T3.2: Extract metadata from XML Content Box.

Internal task T3.3: Design and implement the Java class.

Internal task T3.4: Design and implement the Javascript file.</td><td>Deliverables:
Server part JLINK file viewer</td><td>Dates:
7/03/2022</td></tr>
</table>

Project: Implement front-end JLINK file viewer interface.	**WP ref:** 4	
Major constituent: JLINK viewer.	**Sheet 3 of 4**	
Short description: Once the server part of the viewer is implemented, develop the application interface to make it more attractive and intuitive for the user.	**Planned start date:** 15/04/2022 **Planned end date:** 08/05/2022	
	Start event: **End event:** 13/05/2022	
Internal task T4.1: Work in the user interface. **Internal task T4.2:** Test the JLINK viewer.	**Deliverables:** User interface JLINK file viewer	**Dates:** 08/05/2022

Project: Implement the JLINK files modifier and advanced features.	**WP ref:** 5	
Major constituent: JLINK file modifier.	**Sheet 3 of 4**	
Short description: Implement in the application the ability to modify a JLINK file, either to add a new branch image, or delete it. In addition, advanced features will be implemented, such as the management of JLINK file metadata in a database.	**Planned start date:** 09/05/2022 **Planned end date:** 16/06/2022	
	Start event: 09/05/2022 **End event:** 21/06/2022	
Internal task T5.1: Implement the JLINK files modifier. **Internal task T5.2:** Test JLINK files modifier. **Internal task T5.3:** Implement the advanced features. **Internal task T5.4:** Test the JLINK application.	**Deliverables:** JLINK files modifier JLINK application	**Dates:** 29/05/2022 16/06/2022

Project: Documentation.	**WP ref:** 6	
Major constituent: Activities Review.	**Sheet 4 of 4**	
Short description: Investigate and report all the tasks done during the project and write the degree thesis.	**Planned start date:** 14/02/2022 **Planned end date:** 21/06/2022	
	Start event: 14/02/2022 **End event:** 21/06/2022	
Internal task T6.1: Investigate the project state of the art. **Internal task T6.2:** Write Project Proposal and Work Plan. **Internal task T6.3:** Write Critical Review. **Internal task T6.4:** Write Degree Thesis.	**Deliverables:** Project Proposal and Work Plan Critical Review Degree Thesis	**Dates:** 08/03/2022 14/04/2022 21/06/2022

Project: Defense of the project in court.	**WP ref:** 7	
Major constituent: TFG defense.	**Sheet 4 of 4**	
Short description: Prepare and present the defense of the project in front of an expert tribunal.	**Planned start date:** 22/06/2022 **Planned end date:** 08/07/2022	
	Start event: 22/06/2022 **End event:**	
Internal task T7.1: Elaborate the defense presentation. **Internal task T7.2:** Present the defense in court.	**Deliverables:** Project defense presentation	**Dates:** 04/07/2022 – 08/07/2022

The milestones of the project define reference points that signal an important event. Milestones for each task and their deliverables are shown in the following table:

Table 2. Project milestones

WP#.Task#	Short title	Milestone / deliverable	Date (week)
1.1	Run the reference software	-	4
1.2	Verify if it works in web app	-	4
2.1	Initial design	Proposal System Block Diagram	5
2.2	Implement JLINK file creator	JLINK file creator	6
3.1	Parse the JLINK File	-	7
3.2	Extract metadata from XML Content Box	-	8
3.3	Design and implement the Java class	-	9
3.4	Design and implement the Javascript file	Server part JLINK file viewer	10
4.1	Work in the user interface	-	12
4.2	Test the JLINK viewer	User interface JLINK file viewer	14
5.1	Implement the JLINK files modifier	-	16
5.2	Test JLINK files modifier	JLINK files modifier	17
5.3	Implement the advanced features	.	17
5.4	Test the JLINK application	JLINK application	21
6.1	Investigate the project state of the art	-	5
6.2	Write Project Proposal and Work Plan	Project Proposal and Work Plan	4
6.3	Write Critical Review	Critical Review	9
6.4	Write Degree Thesis	Degree Thesis	19
7.1	Elaborate the defense presentation	Project defense presentation	21
7.2	Present the defense in court	-	21

1.5. Time plan

As mentioned in the previous section, WP3 and WP4 have been carried out in parallel, as shown in the Gantt chart. The implementation tasks T2.2, WP3, WP4, T5.1 and T5.4 have been carried out in a longer period of time than estimated. However, during the test tasks T4.2, T5.2 and T5.4 there were not a large number of bugs to fix, so the time schedule of the project was not affected.

The complete Gantt diagram is presented on the next page:

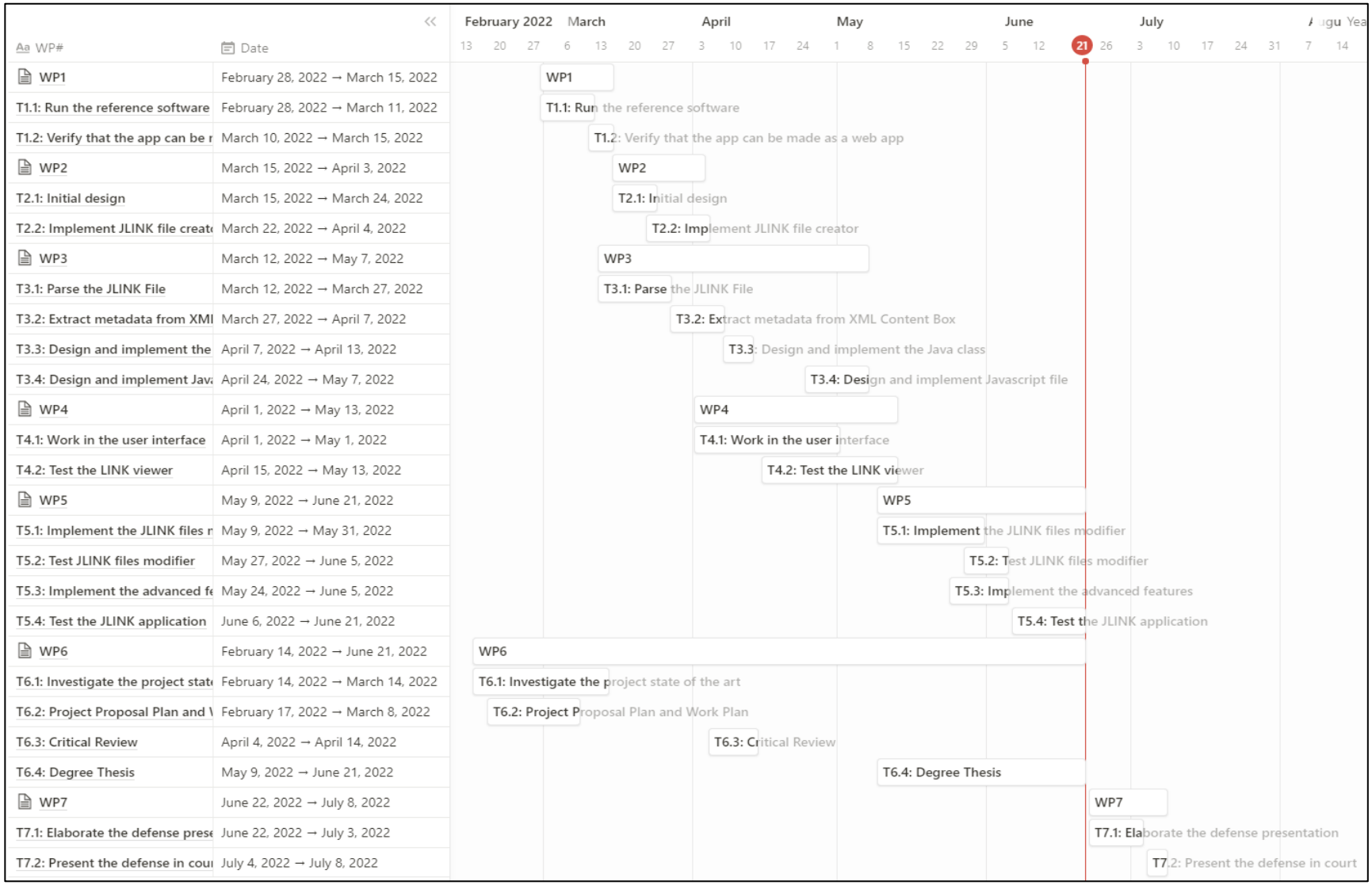


Figure 1. Gantt diagram of project work packages and tasks

2. State of the art

2.1. JPEG Standard

JPEG is an acronym for the “Joint Photographic Experts Group” [1], which introduced the standard in 1992, defines a lossy image compression based on the Discrete Cosine Transform (DCT). The committee has worked on a systematic review and consolidation of its file formats, functionalities and code stream syntax, with the objective to define an overall framework for future and legacy standards called JPEG Systems [2] to ensure interoperability and functionality exchange between the standards, it works as is a multi-part specification, currently consists of eight parts. The relevant parts for this work are explained next.

2.2. Part 5: JPEG Universal Metadata Box Format (JUMBF)

The JPEG Universal Metadata Box Format (JUMBF) [4] is the fifth part of JPEG Systems and provides the rules to embed and refer generic metadata in JPEG files. A box is a binary structure which encapsulates an object. A superbox is a box that may carry other boxes or superboxes. At JUMBF there are three different types of boxes:

- JUMBF Super Box: shall contain exactly one JUMBF Description box and then one or more Content Boxes. The type of the Content Boxes is implied by the JUMBF TYPE field in the JUMBF Description box. JUMBF boxes can be nested.
- JUMBF Description Box: is the first box in the JUMBF superbox, provides additional information about the behaviour and Content of the parent JUMBF box.
- JUMBF Content Box: contains the payload data which is intended to be embedded in a JUMBF Box. In this project the data stored are XML and Codestream Type.

2.3. Part 7: JPEG Linked Media Format (JLINK)

The JPEG Linked Media Format (JLINK) [3] international standard is the seventh part of JPEG Systems, which enables the embodiment of multiple image types and media elements into a single media content. It is implemented using JUMBF boxes, and a new type of JUMBF box is added.

2.3.1. JLINK Concept

The two main parts that make up a JLINK are the 'scene' and the 'link'. The scene shows an image and its metadata, such as the title, the description, and the sprites that indicate the location of the links that come from that scene. A link describes how they are associated and how the transitions between scenes should be, and contains the information of the sprite.

The result of combining scenes and links is the ability to perform a dynamic exploration of several images through sprites, containing all these elements and the information to relate them in the same file.

2.3.2. JLINK User experience

For a typical use of the JLINK, a viewer must provide the following points regarding the user experience, shown in Figure 2:

- Presentation of a scene and sprites: an image corresponding of any scene is presented to a viewport. When other scenes are linked to the presented scene, the sprite corresponding to each link is located at each linkage region on the image of the presented scene. (Step 'a' in Figure 2).
- Selection of a sprite: sprites placed on the scene receive user interface events. (Step 'b' in Figure 2).
- Visual effect for scene change and preparing subsequent scene: when a sprite receives a user interface event, the screen transitions to another scene that is the destination of the corresponding link, the animation of the transition is called 'Jump-In' effect (Transition 'I' at step 'b' in Figure 2). Also, when the back button receives a user interface event, the screen transitions to another scene that is the scene that the presented scene is linked, the animation of the transition is called 'Jump-Out' effect (Transition 'II' at step 'd' in Figure 2). Both animations will be described below.
- Presentation of the destination scene and sprites: the image of the destination scene is presented in the same way as described in the first point. (Final state in steps 'c' and 'd' in Figure 2).
- Presentation of a scene with a title and note: each scene can have a title and a note, describing the image it contains, the viewer software places a text button on the scene that receives user interface events. (Step 'b' in Figure 2).
- Presentation of a title and note: when the text button receives a user event, the text window appears with the scene associated title and note. (Final sate in step 'b' in Figure 2).

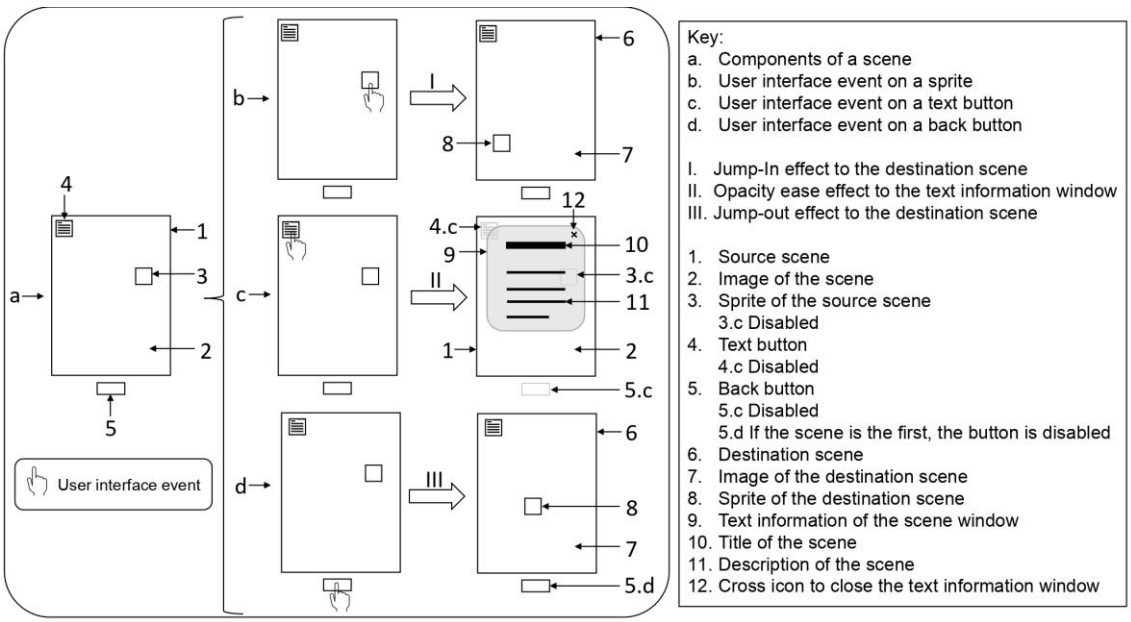


Figure 2. User interface events in a viewer

As mentioned, animated transitions are made while moving between scenes. In this project, those animations have been implemented using HTML5.2 [8], CSS2.1 [9], ECMAScript 2015 [10], and Web Animations [11]:

Jump-In effect is the visual transition for moving forward through a link from the source scene to the destination scene. This effect, shown in Figure 3, proceeds as follows:

1. All the sprites on the source scene disappear and their events are disabled.
2. The destination scene of the jump-in link becomes visible on the source scene in the given viewport, at 0% scale placed at the center point of the corresponding linkage region. (First state in step ‘c’ in Figure 3).
3. The source scene and the destination scene are animated simultaneously.
    i. The source scene is animated to be shown at 300% scale. The source scene fading away to 0% opacity, and shifting the user’s view to put the jump-in linkage point at the center of the viewport. (Step ‘b’ in Figure 3).
    ii. The destination scene is animated to the state to be shown after the effect at scale and position until it fits the display. (Step ‘c’ at in Figure 3).
4. The destination scene sprites appear, and their user interface events are enabled. (Final state in step ‘a’ in Figure 3).

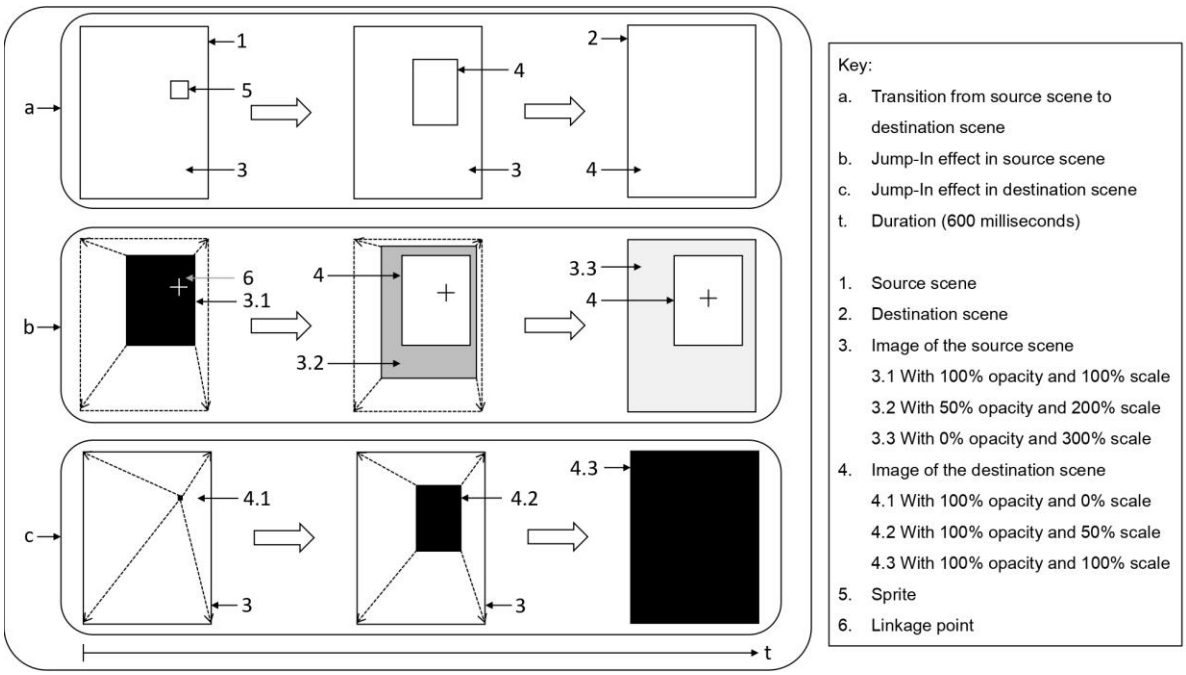


Figure 3: Jump-In effect.

Jump-Out effect is the visual transition for moving backward through a link from the source scene to the destination scene. This effect, shown in Figure 4, proceeds as follows:

1. All the sprites on the source scene disappear and their user interface events are disabled.
2. The destination scene of the jump-out link becomes visible as the background of the source scene in the given viewport, at more than 300% scale scaling value and locating the center of the linkage region at the center of the display. (First state in step ‘c’ in Figure 4).

3. The source scene and the destination scene are animated simultaneously:
   i. The source scene is animated to be shown at 0% scale, locating to the center point of the corresponding linkage region. (Step ‘b’ in Figure 4).
   ii. The destination scene is animated to the state to be shown after the effect at scale and position until it fits the display. (Step ‘c’ in Figure 4).
4. The source scene becomes invisible, and the destination scene sprites appear, and their user interface events are enabled. (Final state in step ‘a’ in Figure 4).

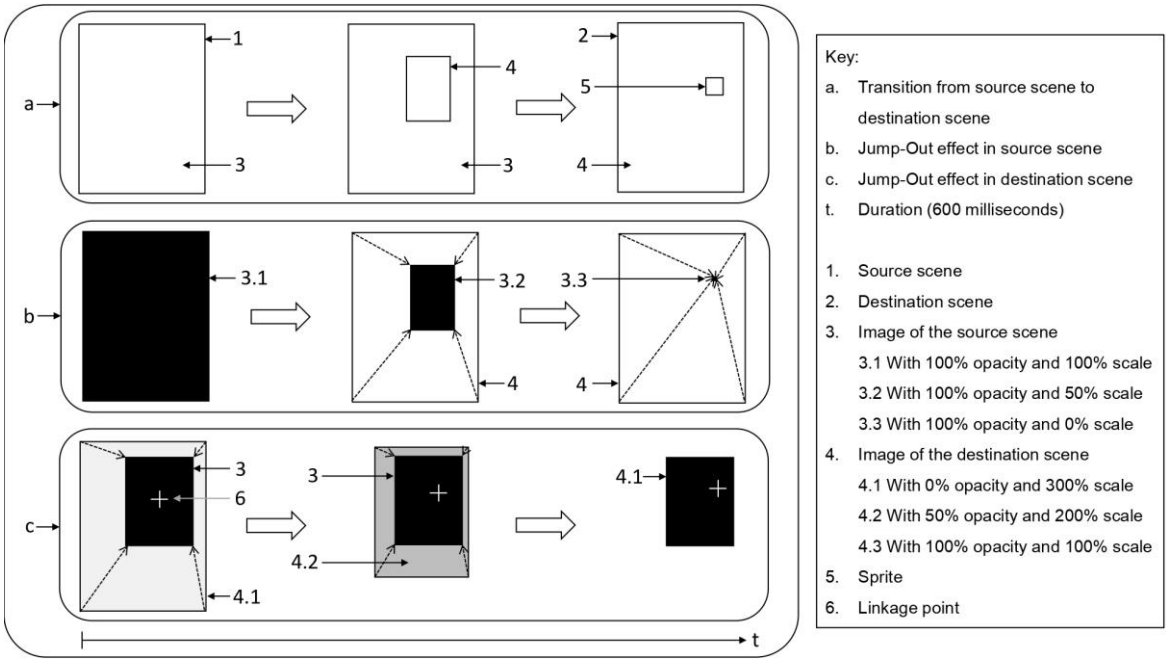


Figure 4. Jump-Out effect

2.3.3. JLINK Components

The JLINK components are the necessary data to be able to interconnect and add more information to the scenes. The components are:

- Version: indicates the version of the scene. In the project, when creating a file, the default string value ("1.0.0") is assigned, and if the file is subsequently modified, the version will increase its value following the specification by semantic versioning [12].
- Title: a string referring the title of the scene. This field is entered by the user.
- Note: a string providing a description text about the scene. This field is entered by the user.
- Viewport: consists of two spatial real coordinates which indicates the centre of the region as a percent ratio for the entire image, two field of view span values as a percent ratio for the entire image and an integer identifier which allows a viewer software to differentiate between viewports. If no viewport is defined, the entire image is used as the viewport with ID equal to 0. No viewports are used in this project, so the entire image is displayed in the viewer.

- Image: contains information about the image displayed in the scene, as the format, indicated by the Media Type of the image, and the Href, that indicates the JUMBF URI reference to the codestream of this image. For the project, the stored images are of type 'image/jpeg'.
- Link: indicates how the current scene is connected to its source. A link is composed of a region, a duration, a viewport ID, a sprite and the source scene which that is connected to.
  - Region: indicates the position of the sprite that the viewer places in the source scene. A linkage region has a shape such as point, rectangle, and ellipse. The location of the linkage region is declared with two spatial real coordinates which indicates the centre of the region as a percent ratio for the entire image. If the shape is a rectangle or an ellipse, the width and height must be indicated as the horizontal and vertical coordinate of the centre of the region as a percent ratio for the entire image, and the rotation angle, which equals 0º in the positive direction of X axis and grows in the counterclockwise direction. In the project, by default the region is a point, and the position values are indicated by the user.
  - Duration: determinates the duration of the animation for scene change in milliseconds. The creator implemented in this project defines the default integer value, 600 milliseconds.
  - VPID: the viewport ID indicates which viewport is to be used in the destination scene by its ID. Since no additional viewports are used, it is assigned the integer value 0.
  - Sprite: declares the JUMBF URI reference to the JUMBF Content Box of the sprite.
  - To: that component declares the JUMBF URI reference to the source scene which the presented image comes from. The user selects the previous scene for each scene.

The mentioned components are declared in the JLINK schema, which is serialized and stored using a subset of the W3C Resource Description Framework (RDF) [13] expressed in XML [14]. It is stored in the file inside an XML JUMBF Content Box.

2.3.4. JLINK Structure

The structure of a JLINK is stored and organized in a JUMBF box structure, as it can be seen in Figure 5. Its structure defines a different approach to JUMBF Super Boxes since JLINK-type JUMBF Super Boxes store multiple JUMBF Super Boxes which are different in type. In a JLINK JUMBF Super Box, it can be found:

- JUMBF Description Box: it is required. Provides additional information about the behaviour and Content of the parent JUMBF box, in that case is the JUMBF JLINK Box. Its JUMBF Type is 0x4C49 4E4B (LINK) and its label indicates what scene the box is storing. In the project, the label to indicate the scenes follows the nomenclature 'scene' plus an incremental integer, starting with zero.
- JUMBF XML Content Box: the scene and link metadata is stored in a XML formatted text file. The file is in turn embedded to the JLINK Box as a JUMBF Super Box with XML type and carried by its own Content Box. This box is required and limited to a single XML box per JLINK box.

- JUMBF JLINK Boxes: the scenes that are linked with a source scene are intended to be embedded within the JLINK box which contains the source scene. Thus, each of these scenes adopt the form of a JLINK type JUMBF Super box and are nested to the parent JLINK box.
- JUMBF Codestream Boxes: both image and sprite from a scene are a picture media type instance. Therefore, JLINK boxes store them as Codestream type JUMBF Super Boxes. In the project, the label to indicate the images and sprites follows the nomenclature 'image' or ‘sprite’ plus an incremental integer, starting with zero.

2.3.5. File position for JLINK Metadata and Linked Media

The structure of a JPEG [15] consists of a sequence of segments, each beginning with a marker that is indicated by beginning with a 0xFF byte and followed by a byte. The essential markers to take into account to locate the JLINK within a JPEG file are:

- Start Of Image (SOI): indicates the beginning of the image.
- Application-specific (APPn): The metadata in JPEG file is stored in APPn. An Exif JPEG file uses an APP1 marker to store metadata.
- Start Of Scan (SOS): Begins a top-to-bottom scan of the image. The next segments is the codestream of the image.
- End Of Image (EOI): indicates the final segment of the image.

JLINK metadata and linked images are included in a broader definition of metadata, as shown in Figure 5. It is possible to signal that the encoded image is located within JLINK Content Type box or is located in the file position of earlier standard. The codestream of the image between the markers SOS and EOI it can be decoded and displayed by a conventional JPEG viewing applications.

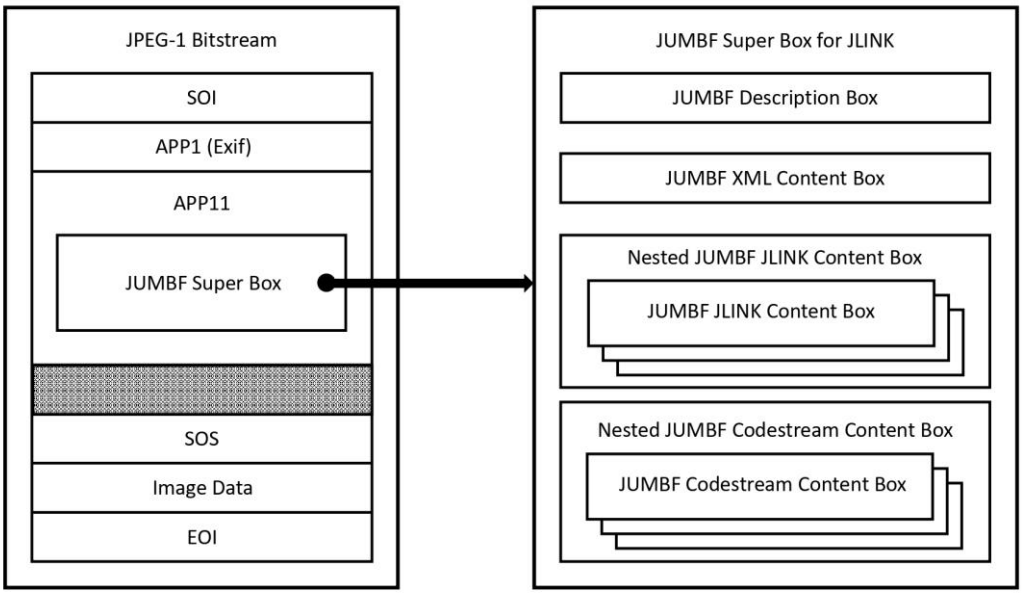


Figure 5. Structure of JUMBF located inside JPEG

3. Project development

The development process of the project has aimed to follow the specifications given by the JLINK standard, resulting in a web application capable of performing various actions on the file defined by the JLINK standard.

3.1. Methodology

Initially, a detailed research process of the JLINK and JUMBF standards was carried out, and the project was also planned, both in terms of the work blocks and the temporal process, as explained in point two of this document. Before starting the implementation, the design of the web pages that make up the application was outlined.

The project was implemented using the integrated development environment Apache NetBeans v13 [16], using the Java SE 8 developer environment [17]. The design of the web application has been done using resources such as Java Server Pages (JSPs) and Servlets [18]. JavaServer Pages [19] are a Sun Microsystems specification for combining Java with HTML to provide dynamic content for Web pages, and a Servlet [20] is a Java programming language class used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. For the development, the web application has been hosted on a GlassFish v5.1.0 server [21], a complete application server that implements the Jakarta EE specification [22]. The project makes use of the code implemented in the thesis JPEG Universal Metadata Box Format and JPEG Linked Media Format: a Reference Software Implementation [6], coded in Java language. Additionally, several java classes have been created to support the actions of the web application. To make the web pages more dynamic and stylish, Javascript scripts [10] were implemented and a Cascading Style Sheets [9] template was applied. The web application also contains a database, connections to the database are made using Structured Query Language (SQL) [23].

The main page of the application, shown at Figure 6, is a menu that gives the user access to the six actions it performs, the implementation and usage of each of them will be detailed below.

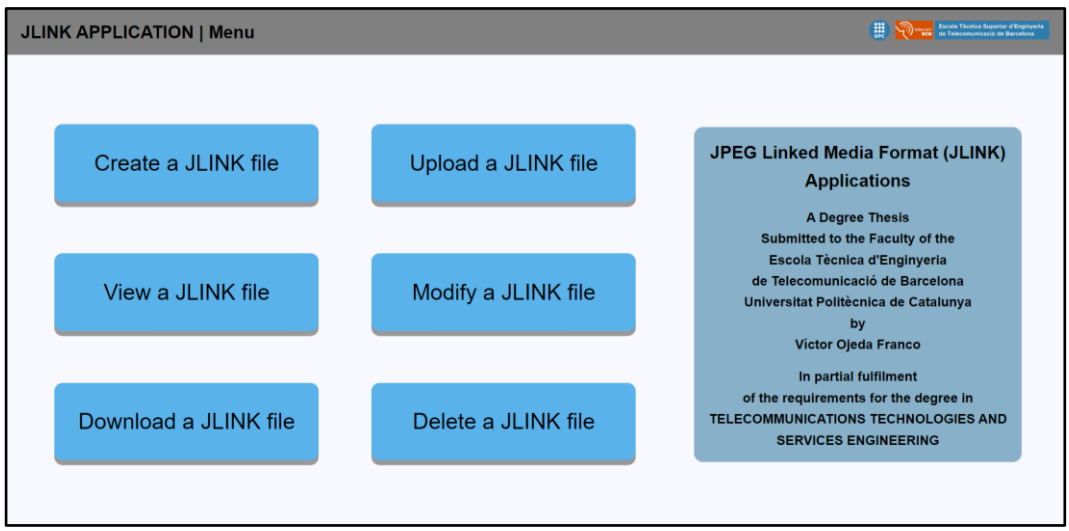


Figure 6. JLINK Application main page

3.2. JLINK File Uploader

This module has the function of uploading JLINK files to the application, to later perform other actions. The process is as follows (The numbering follows the same as the steps in Figure 7):

1. Select the file and add a title and description: the text fields must describe clearly and concisely what is shown in the JLINK file, this information will be saved, together with the upload date, in the database. When uploading the image, it is restricted to be .jpg or .jpeg. When the file is selected, it is displayed on the screen the codestream between the marks SOS and EOI, which is the image that is represented by a common viewer.
2. Upload process: initially, the file is saved in a temporary folder, where it will check that it has been correctly constructed. If so, it will be saved in the save directory of the application and the text data entered on the previous page will be saved in the database. The page shows in a first table the information entered in the database, and then in another table the title, description and the scene from which each scene inside the file comes from. If the user places the mouse over the cells, the image of each scene will be displayed on the screen. The user can return to the main menu or view the file. If the file is badly constructed, it redirects to the error page of the application, which allows to return to the menu.

The file will be well constructed if it follows the structure explained in point 2.3.5 of this document.

To check it, a parse is performed, first the metadata box is traversed, starting with the JUMBF super box dedicated to the main JLINK. The nested JUMBF Codestream boxes are scanned and then the nested JLINK boxes are traversed, and for each of them this action is performed again. If an error occurs, a screen is displayed to the user indicating that an error has occurred while uploading the file, and the option to return to the menu.

The contents of the JUMBF Codestream boxes that belong to the scene images, indicated by their label, are stored in a temporary directory, so that they can be displayed separately on the final screen.

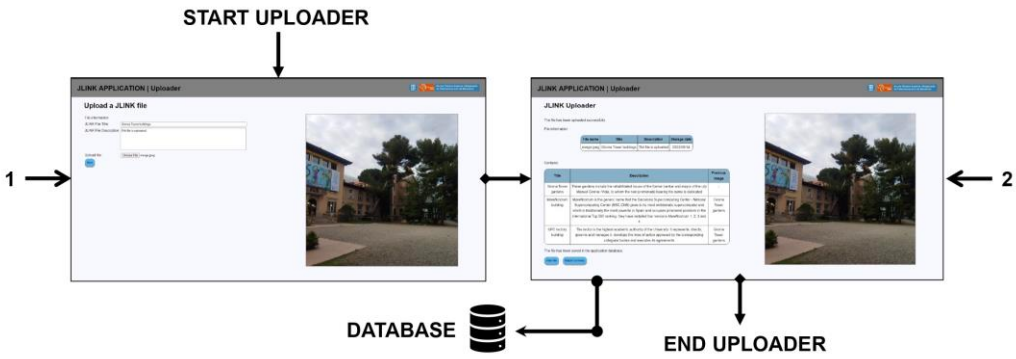


Figure 7. JLINK Uploader in the web application[1]

The uploader has been tested with files created using the demonstrator implemented in the "JPEG Universal Metadata Box Format and JPEG Linked Media Format: a Reference Software Implementation" [6] project, and they have been uploaded and displayed correctly using the web application JLINK viewer.

[1] Appendix A: JLINK Uploader Images

3.3. JLINK File Creator

This module allows the user to create JLINK files, the process is as follows (The numbering follows the same as the steps in Figure 8):

1. Start the creator and enter the main scene information: the user enters the text fields to be saved in the database, and the image, title and description of the main scene (The note component of a scene). When the image is selected, it is displayed on the screen.

At this point initiates the Java class that is used to store the information entered by the user. In order to handle the information interchanged between the different servlets, it is stored as a Servlet context attribute.

2. Add the second scene (I): The page shows in a first table the information entered in the database, and then in another table the title, description and the main scene. If the user places the mouse over the cell, the image of the scene will be displayed on the screen.
3. Add the second scene (II): On the same page as in state two, the user enters the information for the second scene. Through a selector, the user indicates from which scene the scene being added comes from. When the image is selected, it is displayed on the screen.
4. Position the sprite (I): This page shows the image indicated as previous to the one introduced in the previous page. The dimensions of the image are shown and when the mouse is placed over the image it has a crosshair shape. The user cannot click on any of the buttons until the sprite position is selected.
5. Position the sprite (II): when the user clicks on the image, the sprite (created for the application) is displayed above the image, and the coordinates where it is positioned appear on the screen. The user can click again on the image if he wants to change the position of the sprite. The first one allows to add a new scene to the JLINK (moves to step 6) and the other one ends the creation process (moves to step 7).

The information stored is the two spatial coordinates which indicates the center of the region as a percent ratio for the entire image, which is the value of the Link region component, described in section 2.3.3 of this document. The calculation of the location of the sprite and its perceptual value is obtained by an implemented script[2].

6. Adding new scenes: the process is the same as in states 2 and 3. It loops through steps 5, 4 and 3 for each new scene the user adds, until the end button in step 4 is clicked.

During the process of steps 1 to 7, the information about the scenes is stored and grouped in the Java class, which has a tree structure simulating to that of JLINK.

7. File creation: while the file is being built, a loading screen appears. The process may take some time depending on the number of scenes and the dimensions of the images that compose them.

In the construction of the images, the default metadata for each scene that composes the implemented Java class is first completed. Then a JUMBF JLINK Box is created, which is the scene container. In it a JUMBF Description Box is first added with the label assigned to 'scene' plus an integer.

[2] Appendix B: Script that calculates the Link Region of a Scene

Extracting the values from the Java class generates a file in XML format, following the syntax definition indicated by the standard. Then, the file is included inside a JUMBF XML Content Box, and this is inserted in the JUMBF JLINK Content Box corresponding to its scene.

The image is integrated in a JUMBF Codestream Content Box, which is identified with the label with value 'image' plus an integer. For scenes containing nested JUMBF JLINK Content Boxes, a JUMBF Codestream Content Box containing the sprite image is also added, which is identified with label 'sprite' plus an integer. In all three cases the integer is incremental for each box of this type in the file.

Once the JUMBF JLINK Content Boxes have been created for each of the scenes that compose the file, the JLINK structure is assembled. Considering that the structure is linked in the style of a tree, it is formed from the leaves up to the main scene.

Finally, the resulting JUMBF JILNK Content Box of the main scene, with the others linked as appropriate within the nested JLINK Content Box, is merged into the metadata of the JPEG file corresponding to the main scene image, which using a common viewer is displayed.

8. End creator: when the file is finished building, a page similar to the one described in JLINK Uploader step 2 is displayed with the addition of a downloading button. The file is located in the save directory of the application and the text data entered on the previous step 1 is stored in the database.

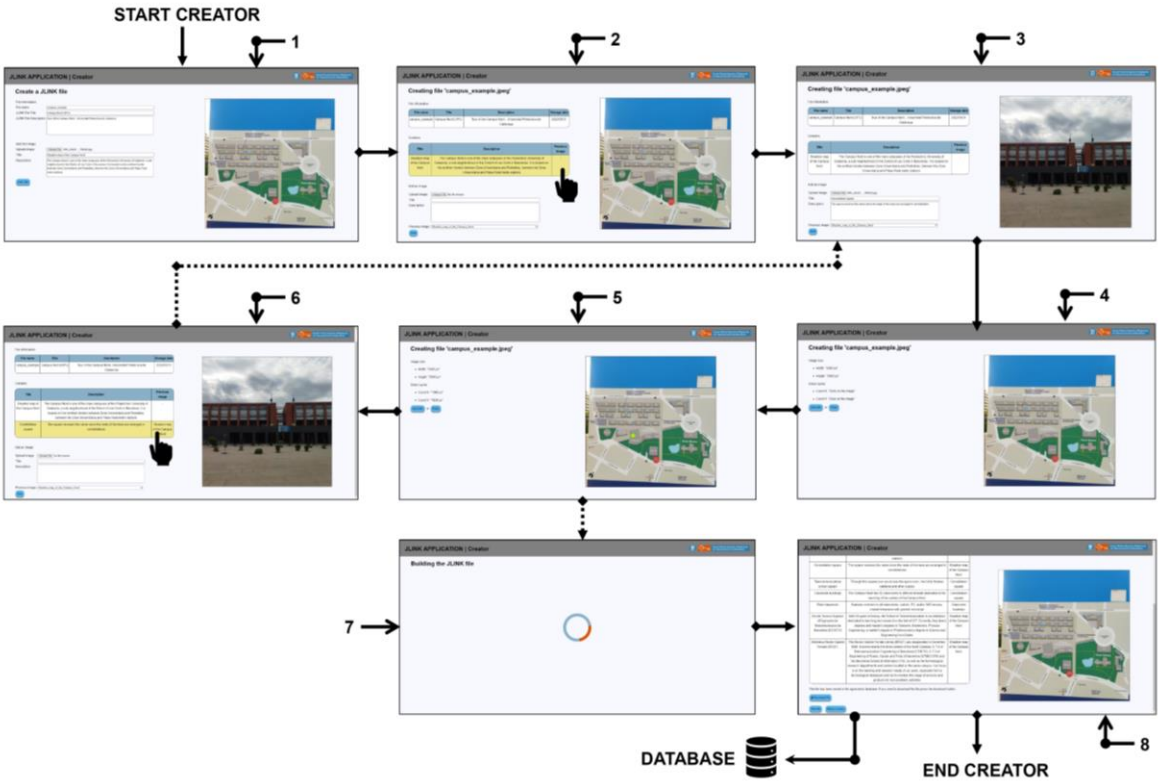


Figure 8. JLINK Creator in the web application[3]

[3] Appendix C: JLINK Creator Images

3.4. JLINK File Viewer

This module accomplishes the main objective of the project, it allows the user to view JLINK files, following the specifications given by the user experience explained in section 2.3.2 of this document. The process is as follows (The numbering follows the same as the steps in Figure 9):

1. List of saved files: first the files stored in the application are displayed in a table, and the information is extracted from the database. The user hovering the mouse over the cells previews the image of the main scene on the screen, and clicking redirects to the page to view the complete file (moves to step 3). Additionally, a search can be performed by file name, file title or file creation/upload date (fields can be excerpts of the text, it does not have to be exactly the same) (moves to step 2).
2. Search result: a page similar to state 1 is displayed, showing the search result in the table.

The file selected by the user is then parsed. The metadata field of the file is accessed, and the first JUMBF JLINK Content Box is traversed. For each of them, the data from the JUMBF XML Content Box is extracted first. As explained in section 2.3.4 of this document, this box is unique within a JUMBF JLINK Content Box. The extracted values are stored in a Java class, which will have the information and structure of the scenes in the file.

Afterwards, the Nested JUMBF Codestream Content Boxes are accessed. For each scene the codestream of the image that composes it is saved in a temporary directory. For scenes that contain links, also save the codestream referring to its sprite. In both cases the name of the saved files is the name of the label of its box.

Finally, access to the Nested JUMBF JLINK Content Boxes, and the same actions explained for each box are performed.

Using the collected information, a Javascript file is formed, stored in the same directory as the images, which describes the following mechanics:

- Initializes the viewer window display. The image of the main scene is adapted to the available size offered by the user's screen. The scene components are positioned.
- Implements the Jump-In effect to run when the user clicks on a sprite.
- Implements the Jump-Out effect to run when the user clicks the back button.
- Implements the display of the textual information of the scene when clicking on the text button.
- Updates the textual values displayed when changing scenes.
- Enables and disables interactive points when required.
- Adjusts the display and position of scene components when the user changes the size of the browser window.

3. Main scene display: the main scene image, the sprites indicating its links, the text icon and the back button (currently disabled) are displayed on the screen.
4. Main scene text information: when the user clicks on the text icon, a window appears above the scene showing the title and description of the scene. The sprites, the text icon and the back button are disabled, and to return to step 2, the close icon must be clicked.

5. Transition to another scene: when the user clicks on a sprite, a transition with Jump-In effect is made to the linked scene. Now the back button is enabled and allows us to go back to the main scene, which is done via the Jump-out effect.
6. Destination scene text information: identical to step 3, for the current scene shown.

Steps 3 to 6 apply for all scenes contained in the JLINK file and are in the same web page. Additionally, the file information stored in the database is displayed, as well as a button to return to step 1 (View another file) and a button to redirect to the menu (Return to menu). If the browser window is moved to another monitor, the viewer pages are resized to fit the new window in order to correctly display the scenes and their components.

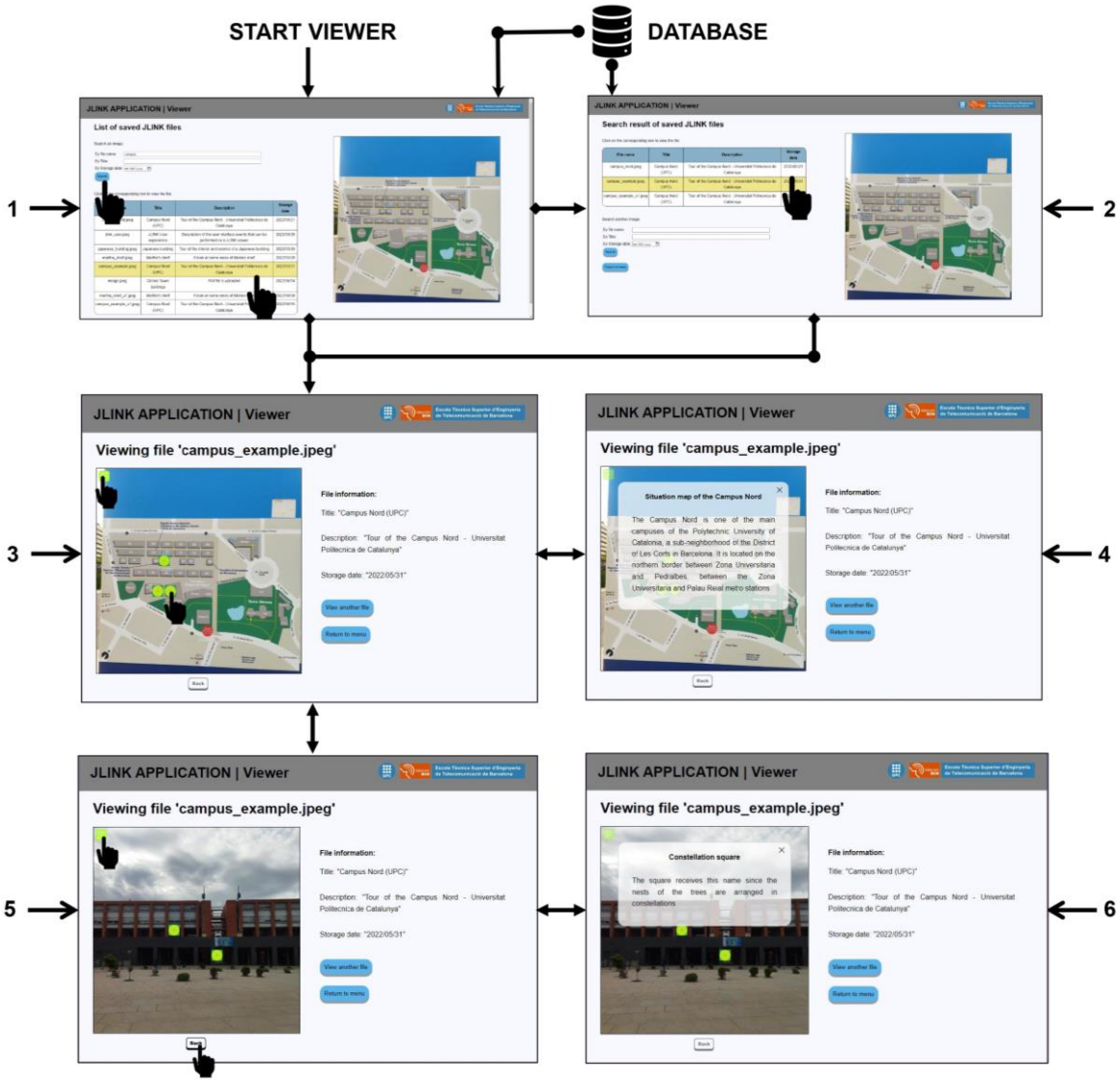


Figure 9. JLINK Viewer in the web application[4]

[4] Appendix D: JLINK Viewer Images

3.5. JLINK File Modifier

This module allows the user to modify the JLINK files that are saved in the database of the application, the process starts with a list and search engine similar to steps 1 and 2 in Figure 9. Clicking on the desired file takes the user to the modifier's main page, where the user will be able to follow these steps (The numbering follows the same as the steps in Figure 10):

1. Modifier main page: the file information is displayed on the screen, followed by a table showing the images of the scenes they contain, with their title and description. In this screen the user can perform three actions: modify, add or delete a scene.
2. Access to the scene modifier: when the user moves the mouse over the rows of the table of scenes contained in the file, the corresponding image will appear on the left side of the screen. By clicking on the row the user can change some scene settings.
3. Scene modifier: in JLINK we can divide scenes into three different types: main scene, scene with links and scene without links. In all three cases the user can change the texts of the title and description of the scene. In case the scene has links (the main scene also has links) you can change the colour of the sprites that appear in that scene to green, blue or red. In case the scene is not the main scene, the user is able to change the duration of the transition between the current scene and the scene it comes from. The value of the current version of the scene is displayed on the screen, which will increase as leaving this page following the specification by semantic versioning [12].

The motivation for adding this enhancement configuration is to be able to explore the options offered by the standard using the implemented viewer.

4. Add a scene: the operation is identical to the creator module.
5. Position the sprite: the process is the same as in steps 4 and 5 of the creator module.
6. Delete a scene: the user can delete scenes that do not contain links. If a scene contains only one link and this link is deleted, in the next action it will be possible to delete this scene.
7. Do another action: at the end of any of the three actions the user can perform other action again or end the process and create the new file.
8. File creation: the process is the same as in step 7 of the creator module.

The file construction process is the same as explained in section 273.3, in this case the modifications given by the user are taken into account.

9. End modifier: the resulting page is as in the creator module in step 8.

The resulting file will have the name of the original file with the addition of '_v' plus an integer identifying which version of the file it is. The integral value is incremented for each modification made to the original file or one of its modifications.

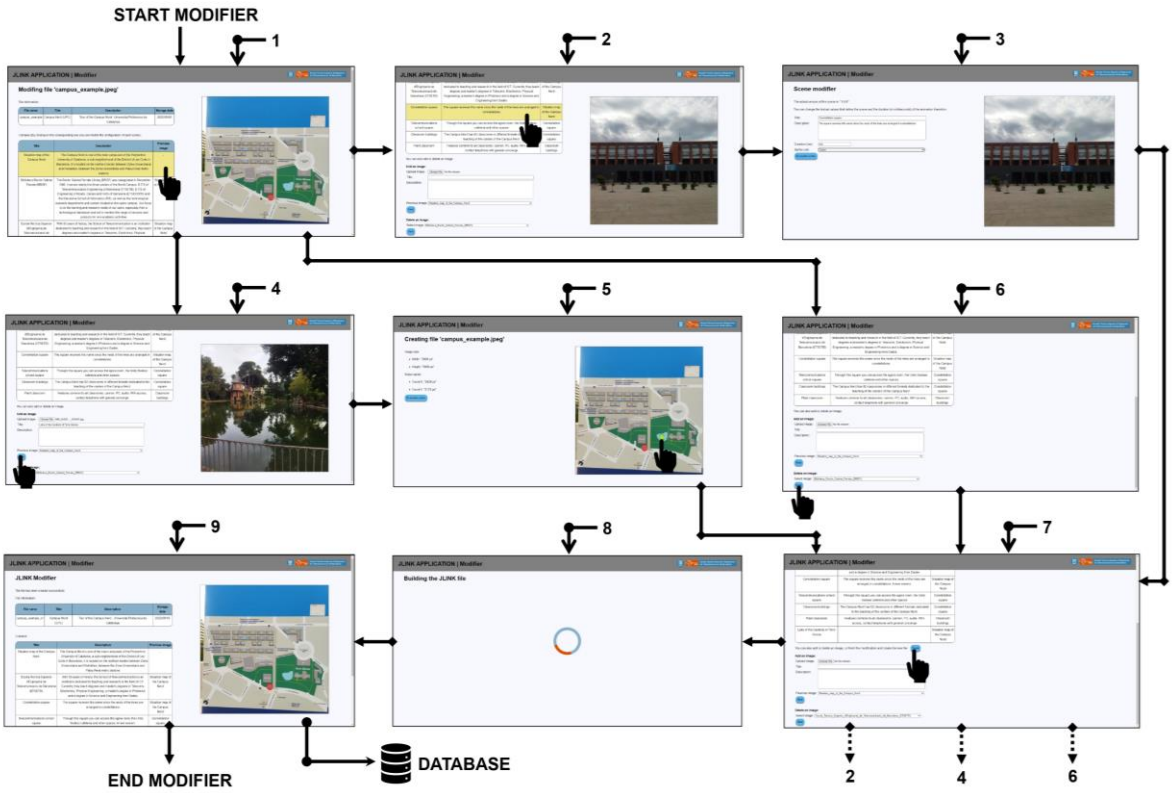


Figure 10. JLINK Modifier in the web application[5]

3.6. Other actions

In addition, the web application also allows downloading or deleting files stored in the database. Both modules consist of a first page which displays all saved images, and a search engine, similar to those in Figure 9 steps 1 and 2.

In the case of the JLINK File Downloader, clicking on the corresponding row in the table downloads the file to the user's download directory, as shown in Figure 11.

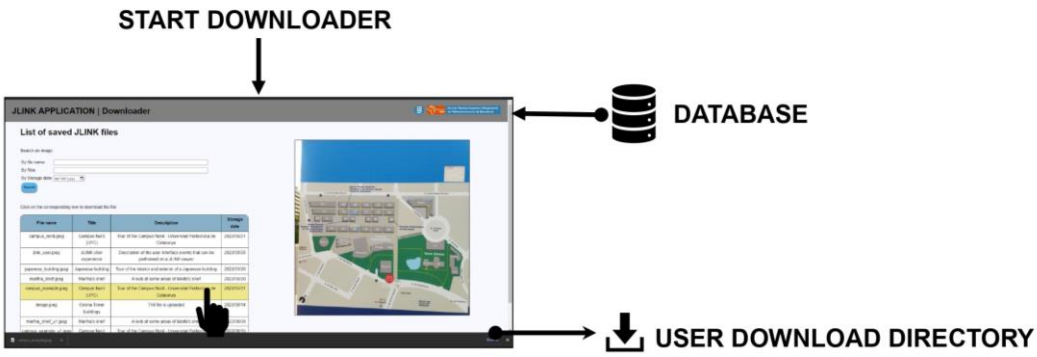


Figure 11. JLINK Downloader in the web application[6]

[5] Appendix E: JLINK Modifier Images
[6] Appendix F: JLINK Downloader Image

For the JLINK File Deleter module, clicking on the corresponding row displays a window in the browser to confirm that the user wants to delete the file. If so, the file is deleted from the application save directory and from the information stored in the database, as shown in Figure 12.

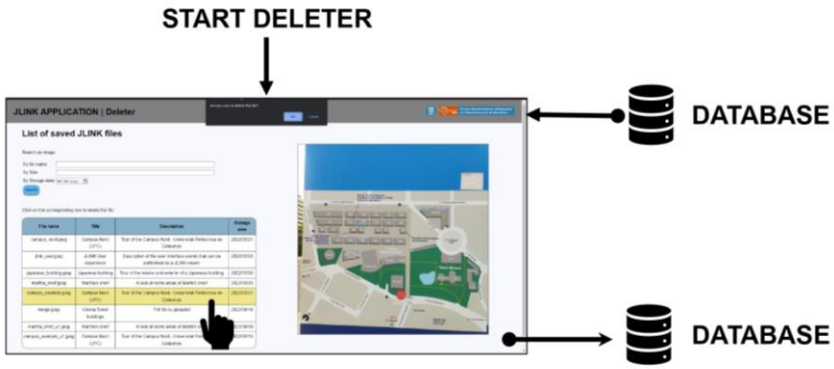


Figure 12. JLINK Deleter in the web application[7]

If an error occurs in any of the application processes, an error page will appear on the screen, indicating to the user whether the error is due to the interaction with the database or when creating or parsing a JLINK file.

[7] Appendix G: JLINK Deleter Image

4. Results

The result of this project has been a web application that meets the objectives stated in section 1.1 of this document, complying with the milestones of the expected deliverables. The final result is that the developed application makes a correct use of the JLINK standard and also presents a neat and clear user interface.

For the creator and modifier modules, the file construction follows the structure defined by the standard, and the declaration of the components of each scene in its metadata field. A loading screen is shown to the user whereas the file is done. On the other hand, the construction time depends on the size of the images of the scenes, and how the set of scenes is structured.

We have found interesting to estimate the approximate building time both for a sequential and parallel structure. It must be taken into account two different scenarios: i) each scene has a link consecutively in a sequential scene and ii) multiple links are contained in the main scene for a parallel structure. Table 3 summarizes the data and Figure 13 and Figure 14 show the file building time with a given number of scenes which image have the same size, for both structures. Note that values have been normalized to the time it takes to build a file with a main scene and a single link. As can be observed in the graphs the trend is linear for a sequential structure and exponential for the parallel one.

Table 3. Temporal progression in the construction of a JLINK file

Linked scenes (number)	Temporal progression for sequential links (a.u.[8] time)	Temporal progression for parallel links (a.u. time)
1	1	1
2	3,55	2,26
3	9,85	5,37
4	24,16	7,95
5	50,17	11,24
6	129,88	14,44

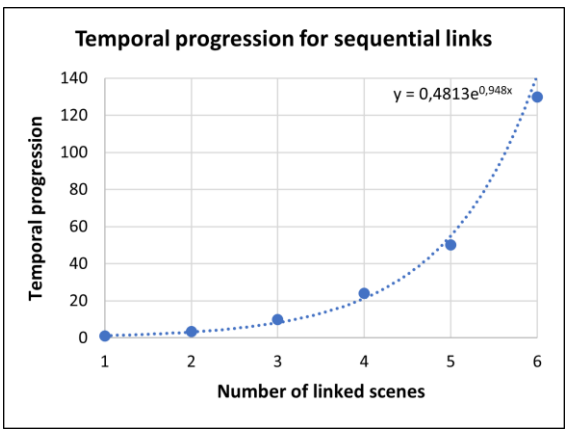


Figure 13. Temporal progression in the construction of a JLINK file for sequential links

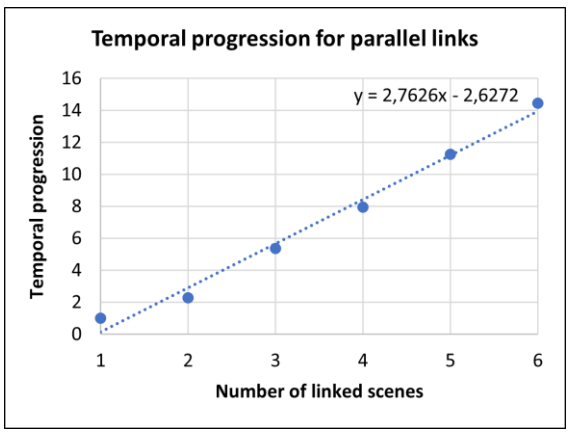


Figure 14. Temporal progression in the construction of a JLINK file for parallel links

[8] a.u.: arbitrary units.

The result obtained in the graphics is as expected, as explained in section 3.3 of this document. Once the JUMBF JLINK Content Box of each scene is built, it is ready to join them according to their structure. The process starts with the scenes without links that are added to their parent scenes, until the main scene is reached. For this reason, the time it takes to encapsulate several JUMBF JLINK Content Boxes in the main box is longer if the links are sequential than if they are parallel.

It should be noted that it is not necessary that all the images in the scenes have the same dimensions, although it is recommended that the images have the same aspect ratio for a better experience when viewing the file.

In the case of the JLINK viewer, the defined objective has been achieved, the development integrates the user experience defined by the standard and offers an intuitive and user-friendly interface to the user. The components of a scene have been correctly displayed and the animations between transitions meet the specifications.

As additional milestones achieved in the project process and that were not part of the objectives stated at the beginning of the project, we have the module that offers the user to upload complete files that follow the JLINK standard. These files could have been created with other JLINK compliant applications or with the one developed in the project.

Furthermore, the web application contains a database, which offers the user a space where to save the information about the stored files and to be able to manage them. In addition, the modules that show the values of the database include a search engine based on different parameters that can be very useful in the day-to-day management of the web.

5. Budget

This section is intended to calculate the budget that would have been cost to realize this project. The project has been carried out by the author alone, but as during the process he has played several roles, the cost calculation will be made taking into account that he has worked in a team with several workers. In a web application development project there are usually the following roles [24]:

- Project manager: this person provides focus for the team, and they helps to keep it moving along on schedule. He manage the budget and takes care of planning. Task T6.2, T6.3, T6.4 and work package 7 are assigned to this role, although its functions are also required throughout the development process.

- Analyst: the role of this person is designed to be a proxy between the client, design, and development. For the project, this role has been carried out in the study of the standards used in the project, in task T6.1.

- Backend developer: this programmer writes code that controls what's displayed on a website. This role has been carried out during implementation tasks T2.2, WP3, T5.1 and T5.3.

- Frontend developer: this person works closely with his backend colleague, the frontend developer takes care of client side: once the page is shown in the browser, it still requires layout to look nice. This role has been carried out during tasks T2.2, T4.1, T5.1 and T5.3.

- Architect: this person configures the infrastructure/hardware on which the websites are running. The role of architect occurred in the initial phase of the project, in WP1.

- UX and UI designer: the UX (user experience) designer is in charge of mapping out the website in a raw sketch so it's clear for the visitors how they can use the website and navigate through it. The UI (user interface) graphic designer creates a visual design for each functionality as a guideline for the Frontend Developer, who will re-create this design by code. For task T2.1, both roles are carried out.

- The QA tester: this person tests the quality assurance (QA) all functions and deliveries that the team has made in every possible condition before the website becomes available to regular users. For test tasks T4.2, T5.2 and T5.4 this role is performed.

Table 4 shows the cost of personnel for the project. In order to calculate the personal fees, it has been taken into account that all roles are junior.

Table 4. Total workers budget calculation

	Salary/hour	Time	Cost
Project manager	29,12 €/h [25]	177 h	5.154,24 €
Analyst	24,79 €/h [26]	54 h	1.338,66 €
Backend developer	20,74 €/h [27]	136 h	2.820,64 €
Frontend developer	21,74 €/h [28]	110 h	2.391,40 €
Architect	23,41 €/h [29]	48 h	1.123,68 €
UX and UI designer	21,89 €/h [30]	16 h	350,24 €
QA Tester	18,47 €/h [31]	80 h	1.477,60 €
		Total cost:	14.656,46 €

We also have to take into account the cost of the material used during the project, in which we have the computer (with its peripherals) and the two documents consulted, which specify the JUMBF and JLINK standards. Furthermore, it is necessary to consider the amortization on the material, which is calculated from the following formula:

$$\text{Amortization} = \frac{\text{Period of use}}{\text{Life cycle}} \cdot \text{Price}$$

Formula 1. Amortization calculation

Table 5 shows the cost of the material used in the project:

Table 5. Total materials budget calculation

	Price	Life cycle	Period of use	Cost
Computer	800 €	4 years	4 months	66,67 €
JUMBF Standard	84 €	4 years	4 months	7,00 €
JLINK Standard	73 €	4 years	4 months	6,08 €
			Total cost:	79,75 €

Since the software used is open source or free of use, we will not have a cost for the project. To the final cost of the project, we will add the electricity and other supplies expenses. The result is shown in Table 6.

Table 6. Total project budget calculation

	Cost
Staff	14.656,46 €
Material	79,75 €
Electricity and other supplies	150 €
Total cost:	14.886,21 €

6. Environment and Social Impact

6.1. Environment Impact

Since the realization of the project and its result does not generate any environmental damage, the impact on nature is minor. The only impact that could be noticed is the electricity that the computer needs to operate.

However, it is necessary to take into account the CO_2 emissions caused by the electricity consumed during the realization of the project and the one that will be caused by a user while using the developed web application.

The energy consumed by the equipment used during the project is calculated by the following formula:

$$\text{Energy consumed [kWh]} = \frac{\text{PC Watts} \cdot \text{ Running Hours}}{1000} = \frac{200\text{W} \cdot 621\text{h}}{1000} = 123{,}2\ \text{kWh}$$

Formula 2. Energy consumed during the project

The Spanish electricity grid mix published by the CNMC on April 20, 2022 is 259 g CO_2eq/kWh [32]. Therefore, the total CO_2 emissions during the project are calculated in the following formula:

$$\text{CO}_2\text{ emissions [kg CO}_2\text{]} = 0{,}259\ \frac{\text{kg CO}_2}{\text{kWh}} \cdot\ 123{,}2\ \text{kWh} = 31{,}9\ \text{kg CO}_2$$

Formula 3. CO_2 emissions during the project

The CO_2 emissions produced while a user is making use of the web application must also be considered. Assuming that on average a user is 25 minutes using the application, the energy consumed is:

$$\text{Energy consumed [kWh]} = \frac{200\text{W} \cdot 25\ \text{min} \cdot \frac{1\text{h}}{60\ \text{min}}}{1000} = 0{,}083\ \text{kWh}$$

Formula 4. Average energy consumed by a user

And the CO_2 emission are:

$$\text{CO}_2\text{ emissions [kg CO}_2\text{]} = 0{,}259\ \frac{\text{kg CO}_2}{\text{kWh}} \cdot\ 0{,}083\ \text{kWh} = 0{,}051\ \text{kg CO}_2$$

Formula 5. Average CO_2 emissions per user

As we can observe from the results obtained, the environmental impact of the project is minimal, although not positive.

6.2. Social Impact

Being able to interact with multiple images offers the user a better experience when viewing 2D content. JLINK provides a wide variety of applications, and through the software implemented in this project, users will be able to make use of it, to build and visualize JLINK files with their ideas. Not only could a common user benefit from the implemented application, but also companies could make use of it.

7. Conclusions and future development:

7.1 Conclusions

In this project a study has been carried out on the specifications of two standards created by the Joint Photographic Experts Group, and their use in a web application in which the possibilities offered by the standards are explored.

Thanks to the overall result of the modules offered by the developed application, it has been understood the great variability of options that can add new features to image files, and how positive it is to store this type of information in their metadata.

Several applications have been found where the use of the standard that defines JLINK can be used, and therefore be explored through the result of this thesis.

In addition, for the implementation of the JLINK viewer, it has been found that some information components are not strictly necessary and could be modified to offer more variety when displaying a scene. For example, the 'To' component of a link that is defined in the XML Content Box for each scene is not necessary, since by the file structure itself it is possible to identify from which scene the current scene comes from. The value of this component could be changed by the JUMBF URI reference of the JUMBF Codestream Box where the sprite is located in the source scene, and the source scene could have stored several different types of sprite, and remove the link sprite component from the source scene.

As far as the development in this thesis is concerned, the author has learned the steps to implement a web application and has gained experience in working with different programming languages. He has also understood the importance of teamwork in projects of this type, in order to be able to perform several tasks simultaneously, since it was not possible to do it in this way during the project process.

7.2 Future development

As a future development it would be possible to realize the following aspects:

- Add a user login in the application, in which the person is able to log in as different profiles. For example, as an administrator, which would be able to create, modify and delete files, and another profile as a user, in which would have access to view and download files from the database.
- Update the software to make it functional to more diversity of devices, a version either for Android or for iOS.
- Finally, if there is access to new versions of the JLINK specification, it will be possible to use the code developed in this project as a basis for new applications offered by the standard.

References:

[1] About Joint Photographic Experts Group (JPEG) committee. [Online] Available: https://jpeg.org/about.html [Accessed: 16 May 2022].

[2] Overview of JPEG Systems. [Online] Available: https://jpeg.org/jpegsystems/index.html [Accessed: 16 May 2022].

[3] JPEG Systems — Part 7: JPEG Linked Media Format (JLINK). ISO/IEC 19566-7:2021 JPEG Systems Part 7: JLINK. [Online] Available: https://www.iso.org/standard/78466.html [Accessed: 14 February 2022].

[4] JPEG Systems — Part 5: JPEG Universal Metadata Box Format (JUMBF). ISO/IEC 19566-5:2018. [Online] Available: https://www.iso.org/standard/73604.html [Accessed: 14 February 2022].

[5] Distributed Multimedia Applications Group. [Online] Available: https://dmag.ac.upc.edu/ [Accessed: 14 February 2022].

[6] JPEG Universal Metadata Box Format and JPEG Linked Media Format: a Reference Software Implementation — A Degree Thesis Submitted to the Faculty of the Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona Universitat Politècnica de Catalunya by Carlos Valverde Díaz. [Online] Available: https://upcommons.upc.edu/handle/2117/368136 [Accessed: 15 June 2022].

[7] Notion.so. [Online] Available: https://www.notion.so/ [Accessed 25 February 2022].

[8] HTML Standard Specification. [Online] Available: https://html.spec.whatwg.org/print.pdf [Accessed: 30 May 2022].

[9] Cascading Style Sheets (CSS) Specification. [Online] Available: https://www.w3.org/TR/css-2021/ [Accessed: 30 May 2022].

[10] ECMAScript® 2021 Language Specification [Online] Available: https://www.ecma-international.org/publications-and-standards/standards/ecma-262/ [Accessed: 30 May 2022].

[11] Web Animations Specification. [Online] Available: https://www.w3.org/TR/web-animations-1/ [Accessed: 30 May 2022].

[12] Semantic Versioning Specification (SemVer). [Online] Available: https://semver.org/spec/v1.0.0.html [Accessed: 30 May 2022].

[13] Resource Description Framework (RDF). [Online] Available: https://www.w3.org/RDF/ [Accessed: 30 May 2022].

[14] Extensible Markup Language (XML). [Online] Available: https://www.w3.org/XML/ [Accessed: 30 May 2022].

[15] JPEG File Structure. [Online] Available: https://docs.fileformat.com/image/jpeg/ [Accessed: 23 May 2022].

[16] Apache NetBeans. [Online] Available: https://netbeans.apache.org/ [Accessed: 20 February 2022].

[17] Java SE 8. [Online] Available: https://www.oracle.com/es/java/technologies/javase/javase8-archive-downloads.html [Accessed: 20 February 2022].

[18] Fusion Middleware Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server. [Online] Available: https://docs.oracle.com/cd/E15523_01/web.1111/e13712/basics.htm#WBAPP117 [Accessed: 02 June 2022].

[19] Overview of Java Server Pages (JSP). [Online] Available: https://docs.oracle.com/cd/E13222_01/wls/docs81/jsp/intro.html [Accessed: 02 June 2022].

[20] Java Servlet Technology. [Online] Available: https://docs.oracle.com/javaee/7/tutorial/servlets.htm [Accessed: 02 June 2022].

[21] Eclipse GlassFish. [Online] Available: https://projects.eclipse.org/projects/ee4j.glassfish [Accessed: 20 February 2022].

[22] Jakarta EE Specifications. [Online] Available: https://jakarta.ee/specifications/ [Accessed: 02 June 2022].

[23] Web SQL Database. [Online] Available: https://www.w3.org/TR/webdatabase/ [Accessed: 02 June 2022].

[24] 7 Essential Roles in a Web Development Team. [Online] Available: https://blog.acolad.com/7-essential-roles-in-a-web-development-team [Accessed: 12 June 2022].

[25] Salaries for Junior Project Manager in Spain. [Online] Available: https://www.glassdoor.es/Sueldos/project-manager-junior-sueldo-SRCH_KO0,22.htm [Accessed: 12 June 2022].

[26] Salaries for Junior Analyst in Spain. [Online] Available: https://www.glassdoor.es/Sueldos/junior-analyst-sueldo-SRCH_KO0,14.htm [Accessed: 12 June 2022].

[27] Salaries for Junior Backend Developer in Spain. [Online] Available: https://www.glassdoor.es/Sueldos/junior-backend-developer-sueldo-SRCH_KO0,24.htm [Accessed: 12 June 2022].

[28] Salaries for Junior Front End Developer in Spain. [Online] Available: https://www.glassdoor.es/Sueldos/junior-front-end-developer-sueldo-SRCH_KO0,26.htm [Accessed: 12 June 2022].

[29] Salaries for the position of Junior Architect in Spain. [Online] Available: https://www.glassdoor.es/Sueldos/junior-architect-sueldo-SRCH_KO0,16.htm [Accessed: 12 June 2022].

[30] Salaries for Junior UX/UI Designer in Spain. [Online] Available: https://www.glassdoor.es/Sueldos/junior-ux-ui-designer-sueldo-SRCH_KO0,21.htm [Accessed: 12 June 2022].

[31] Salaries for Junior QA Tester in Spain. [Online] Available: https://www.glassdoor.es/Sueldos/junior-qa-tester-sueldo-SRCH_KO0,16.htm# [Accessed: 12 June 2022].

[32] Electric power emission factor: the electricity mix. [Online] Available: https://canviclimatic.gencat.cat/es/actua/factors_demissio_associats_a_lenergia/ [Accessed: 12 June 2022].

Appendix A: JLINK Uploader Images

This appendix shows in detail the screenshots of Figure 7.

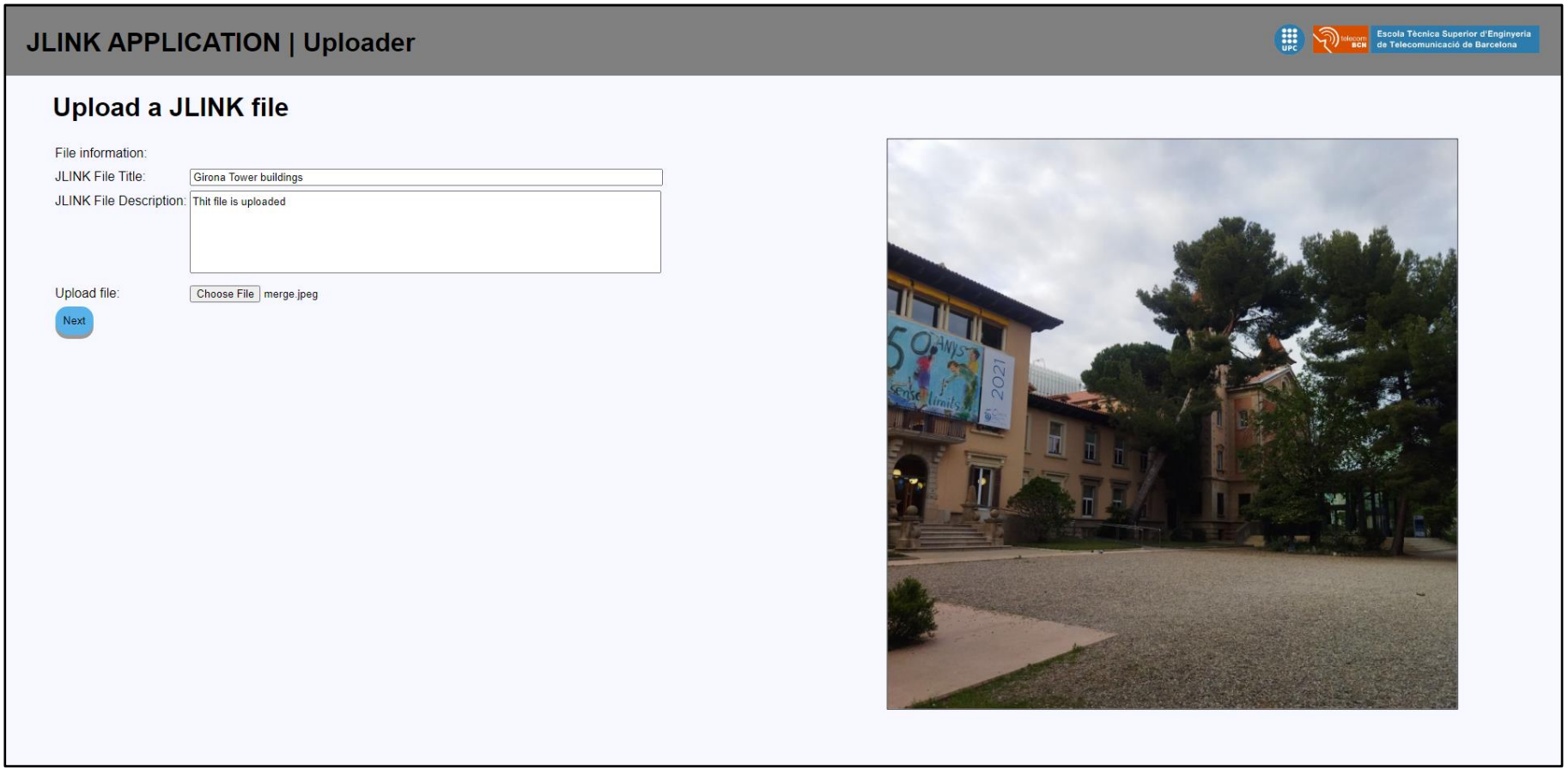


Figure A1. Image 1 from Figure 7.

JLINK APPLICATION | Uploader

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

The file has been uploaded successfully.

File information:

File name	Title	Description	Storage date
merge.jpeg	Buildings in Girona Tower gardens	Tour of the Girona Tower gardens and a look of the two main buildings	2022/06/01

Contains:

Title	Description	Previous image
Girona Tower gardens	These gardens include the rehabilitated house of the former banker and mayor of the city Manuel Girona i Vidal, to whom the next promenade bearing his name is dedicated	-
MareNostrum building	MareNostrum is the generic name that the Barcelona Supercomputing Center - National Supercomputing Center (BSC-CNS) gives to its most emblematic supercomputer and which is traditionally the most powerful in Spain and occupies prominent positions in the international Top 500 ranking. they have installed four versions MareNostrum 1, 2, 3 and 4	Girona Tower gardens
UPC rectory building	The rector is the highest academic authority of the University. It represents, directs, governs and manages it, develops the lines of action approved by the corresponding collegiate bodies and executes its agreements	Girona Tower gardens

The file has been saved in the application database. If you want to download the file press the download button.

Download File

View file | Return to menu

Figure A2. Image 2 from Figure 7.

Appendix B: Script that calculates the Link Region of a Scene

This script calculates the value of the coordinates where the user clicks and the link region component that is indicated in the scene metadata. There is also the function that obtains the original size of the image.

The element obtained by its id "img" is the image of the scene to which the new scene being added by the user is linked.

The element obtained by its id "output_x/y" are the coordinates where the sprite is located.

The element obtained by its id "sprite_x/y" is the value of the link region component.

The element obtained by its id "sprite" is the image of the sprite displayed on the screen.

```
var img = document.getElementById("img");

img.addEventListener("click", function (e) {
    var original_width  = img.naturalWidth;
    var original_height = img.naturalHeight;
    img.x = img.getBoundingClientRect().left;
    img.y = img.getBoundingClientRect().top;
    var result_x = Math.trunc((e.clientX - img.x)*original_width/img.width);
    var result_y = Math.trunc((e.clientY - img.y)*original_height/img.height);
    if (result_x < 0){
        result_x = 0;
    } else if (result_x > original_width){
        result_x = original_width;
    }
    if (result_y < 0){
        result_y = 0;
    } else if (result_y > original_height){
        result_y = original_height;
    }
    document.getElementById("output_x").innerHTML = result_x + ' px';
    document.getElementById("output_y").innerHTML = result_y + ' px';
    document.getElementById("sprite_x").value =
            Math.trunc(result_x*100/original_width);
    document.getElementById("sprite_y").value =
            Math.trunc(result_y*100/original_height);
    document.getElementById("sprite").style.left = e.clientX +
            document.body.scrollLeft +
            document.documentElement.scrollLeft + 'px';
    document.getElementById("sprite").style.top = e.clientY +
            document.body.scrollTop +
            document.documentElement.scrollTop + 'px';
    document.getElementById("sprite").style.display = "block";
    document.getElementById("selectSprite_button").style.pointerEvents = "auto";
});

function getImgSize() {
    var original_width  = img.naturalWidth;
    var original_height = img.naturalHeight;
    document.getElementById("img_width").innerHTML = original_width + ' px';
    document.getElementById("img_height").innerHTML = original_height + ' px';
}
```

Appendix C: JLINK Creator Images

This appendix shows in detail the screenshots of Figure 8.

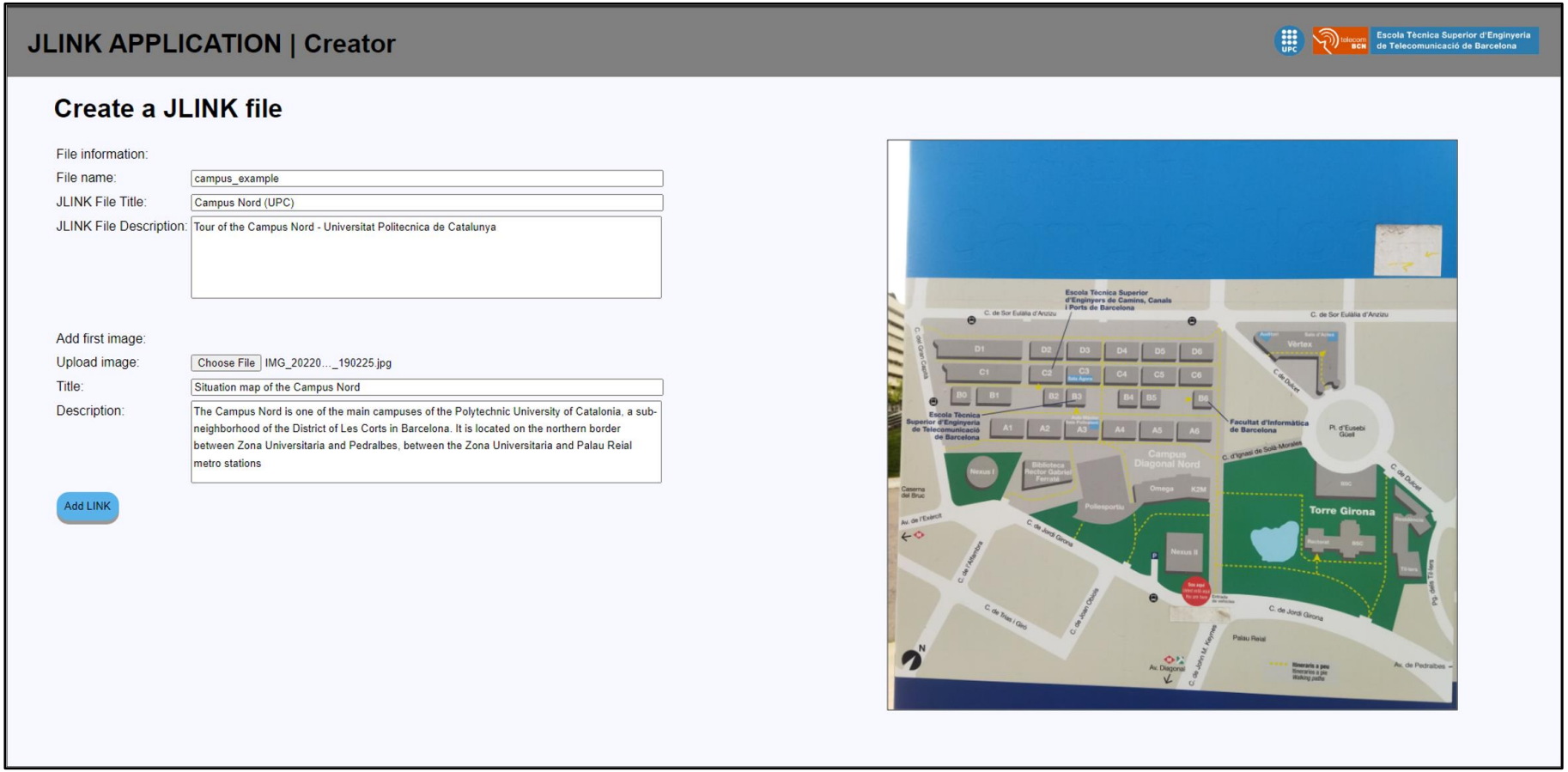


Figure C1. Image 1 from Figure 8.

JLINK APPLICATION | Creator

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Creating file 'campus_example.jpeg'

File information:

File name	Title	Description	Storage date
campus_example	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/05/31

Contains:

Title	Description	Previous image
Situation map of the Campus Nord	The Campus Nord is one of the main campuses of the Polytechnic University of Catalonia, a sub-neighborhood of the District of Les Corts in Barcelona. It is located on the northern border between Zona Universitaria and Pedralbes, between the Zona Universitaria and Palau Reial metro stations	-

Add an image:

Upload image: Choose File No file chosen

Title:

Description:

Previous image: Situation_map_of_the_Campus_Nord

Next

Figure C2. Image 2 from Figure 8.

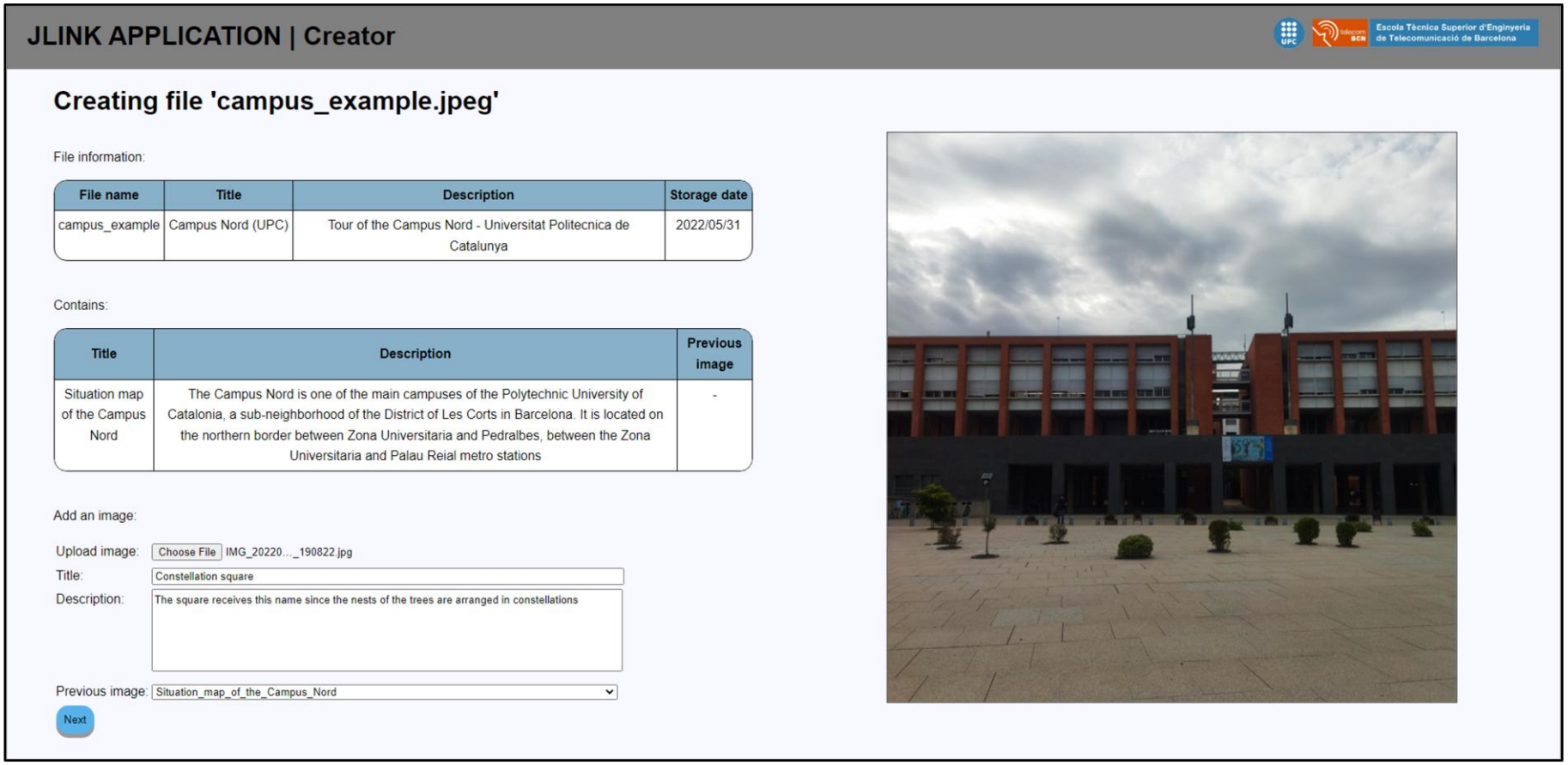


Figure C3. Image 3 from Figure 8.

JLINK APPLICATION | Creator

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Creating file 'campus_example.jpeg'

Image size:

- Width: "3000 px"
- Height: "3000 px"

Select sprite:

- Coord X: "Click on the image"
- Coord Y: "Click on the image"

Add LINK or Finish

Figure C4. Image 4 from Figure 8.

JLINK APPLICATION | Creator

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Creating file 'campus_example.jpeg'

Image size:

- Width: "3000 px"
- Height: "3000 px"

Select sprite:

- Coord X: "1096 px"
- Coord Y: "1826 px"

Add LINK or Finish

Figure C5. Image 5 from Figure 8.

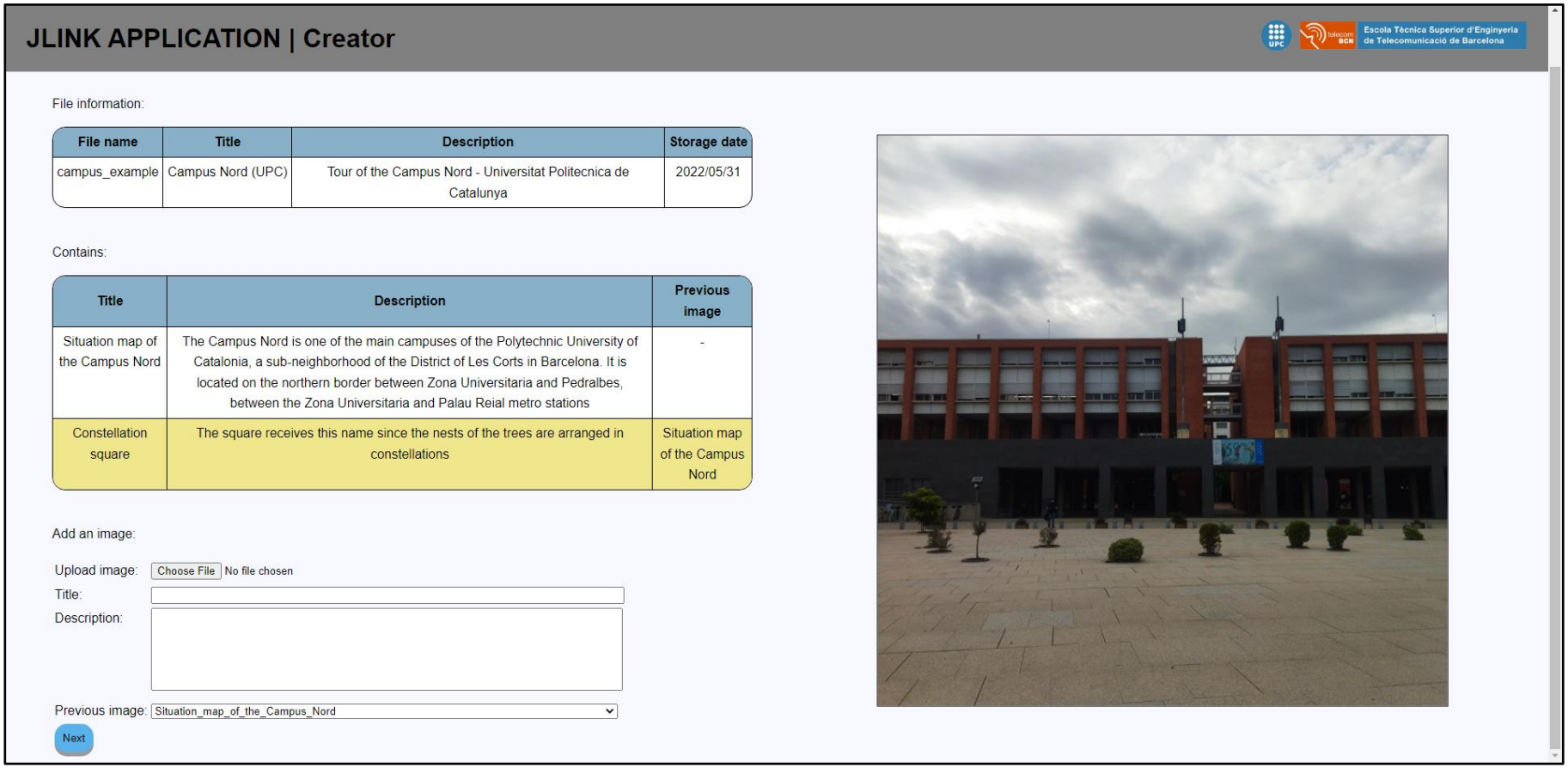


Figure C6. Image 6 from Figure 8.

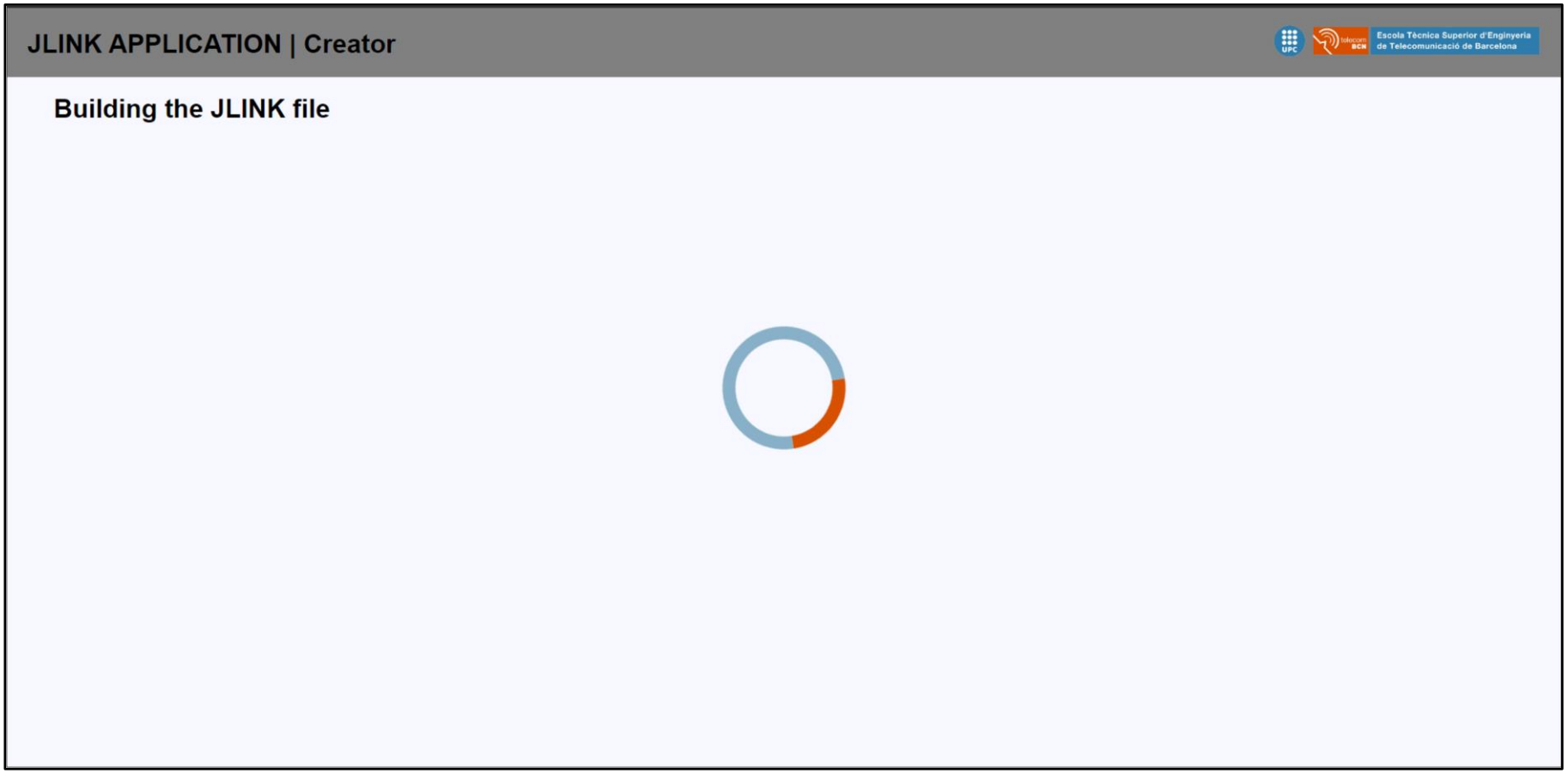


Figure C7. Image 7 from Figure 8.

JLINK APPLICATION | Creator

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

	stations	
Constellation square	The square receives this name since the nests of the trees are arranged in constellations	Situation map of the Campus Nord
Telecommunications school square	Through this square you can access the agora room, the Unity Sodexo cafeteria and other spaces	Constellation square
Classroom buildings	The Campus Nord has 82 classrooms in different formats dedicated to the teaching of the centers of the Campus Nord	Constellation square
Plant classroom	Features common to all classrooms: cannon, PC, audio, WiFi access, contact telephone with general concierge	Classroom buildings
Escola Tecnica Superior d'Enginyeria de Telecomunicacio de Barcelona (ETSETB)	With 50 years of history, the School of Telecommunication is an institution dedicated to teaching and research in the field of ICT. Currently, they teach degrees and master's degrees in Telecoms, Electronics, Physical Engineering, a master's degree in Photonics and a degree in Science and Engineering from Dades	Situation map of the Campus Nord
Biblioteca Rector Gabriel Ferrate (BRGF)	The Rector Gabriel Ferrate Library (BRGF) was inaugurated in December 1996. It serves mainly the three centers of the North Campus: E.T.S of Telecommunication Engineering of Barcelona (ETSETB), E.T.S of Engineering of Roads, Canals and Ports of Barcelona (ETSECCPB) and the Barcelona School of Informatics (FIB), as well as the technological research departments and centers located on the same campus. Our focus is on the learning and research needs of our users, especially from a technological standpoint and not to mention the range of services and products for non-academic activities	Situation map of the Campus Nord

The file has been saved in the application database. If you want to download the file press the download button.

Download File

View file | Return to menu

Figure C7. Image 8 from Figure 8.

Appendix D: JLINK Viewer Images

This appendix shows in detail the screenshots of Figure 9.

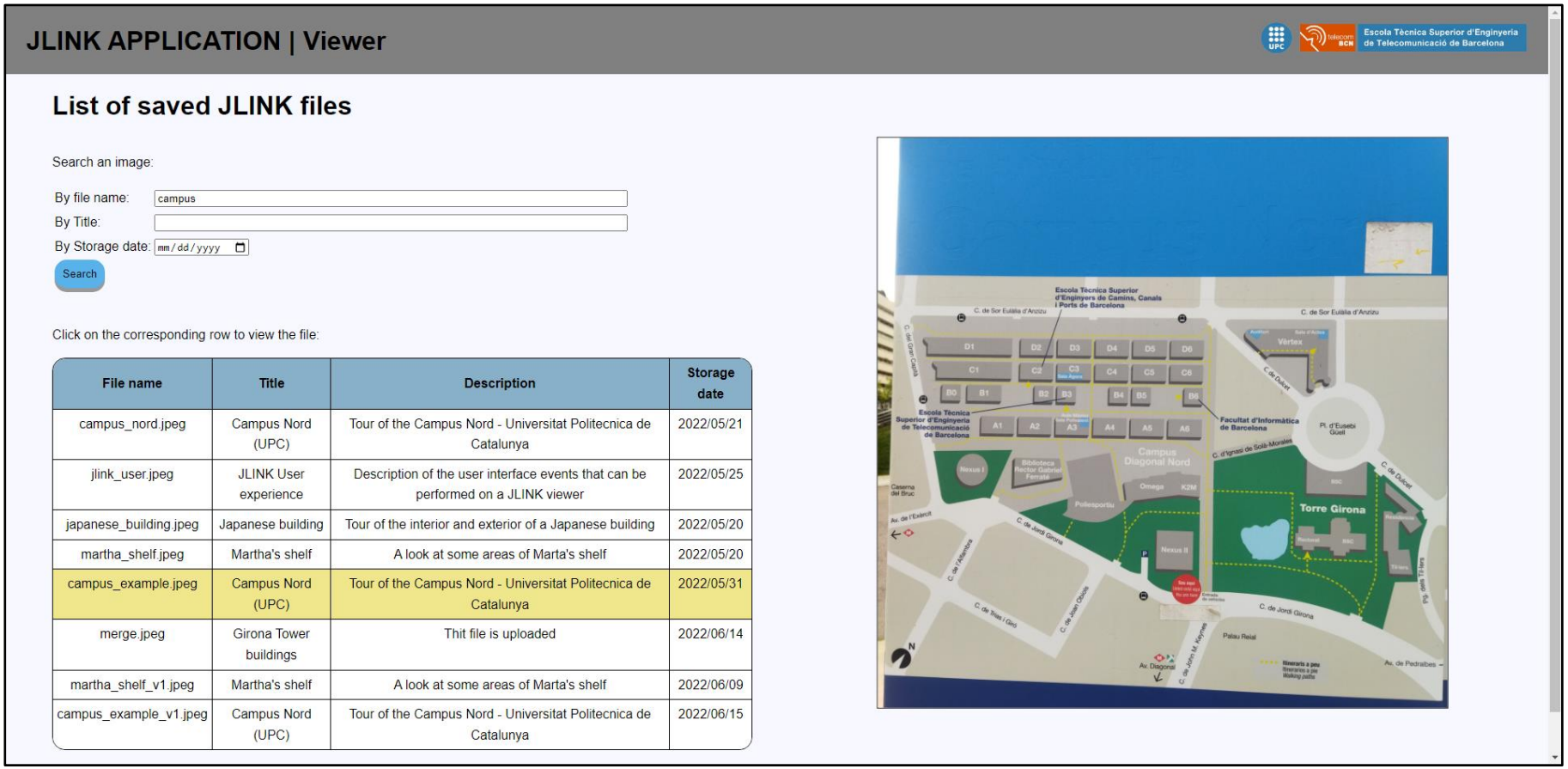


Figure D1. Image 1 from Figure 9.

JLINK APPLICATION | Viewer

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Search result of saved JLINK files

Click on the corresponding row to view the file:

File name	Title	Description	Storage date
campus_nord.jpeg	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/05/21
campus_example.jpeg	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/05/31
campus_example_v1.jpeg	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/06/15

Search another image:

By file name:

By Title:

By Storage date: mm/dd/yyyy

Search

Return to menu

Figure D2. Image 2 from Figure 9.

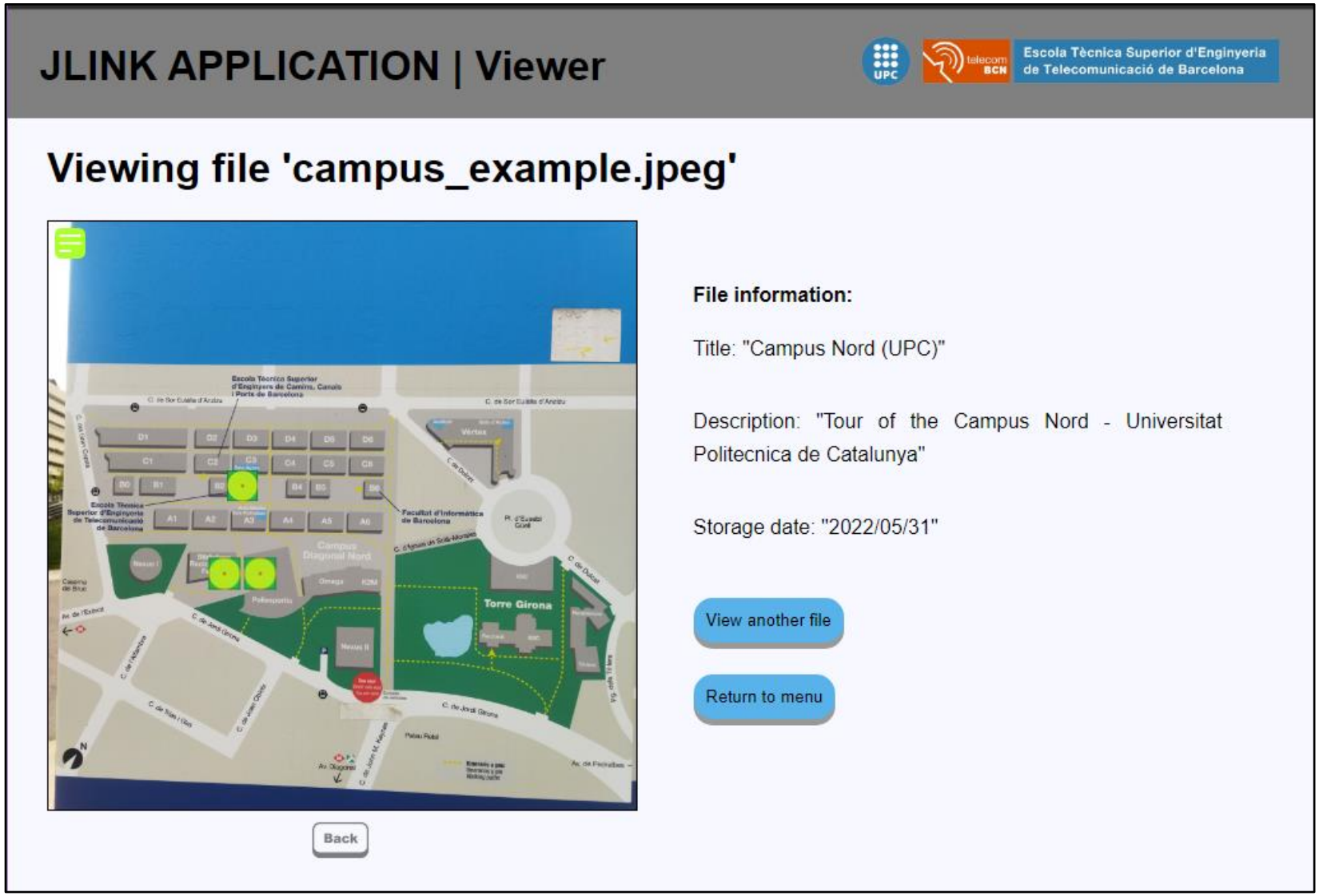


Figure D3. Image 3 from Figure 9.

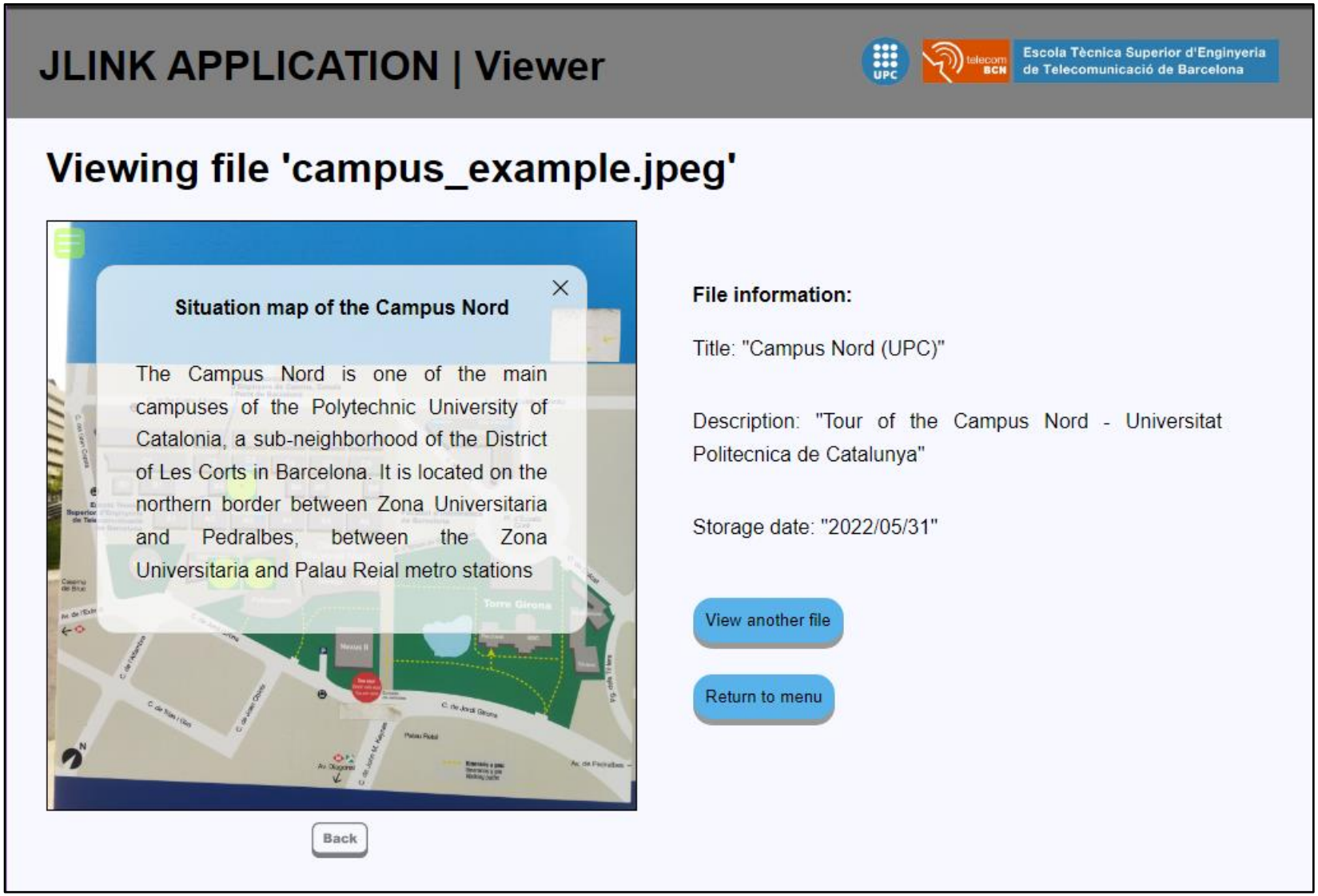


Figure D4. Image 4 from Figure 9.

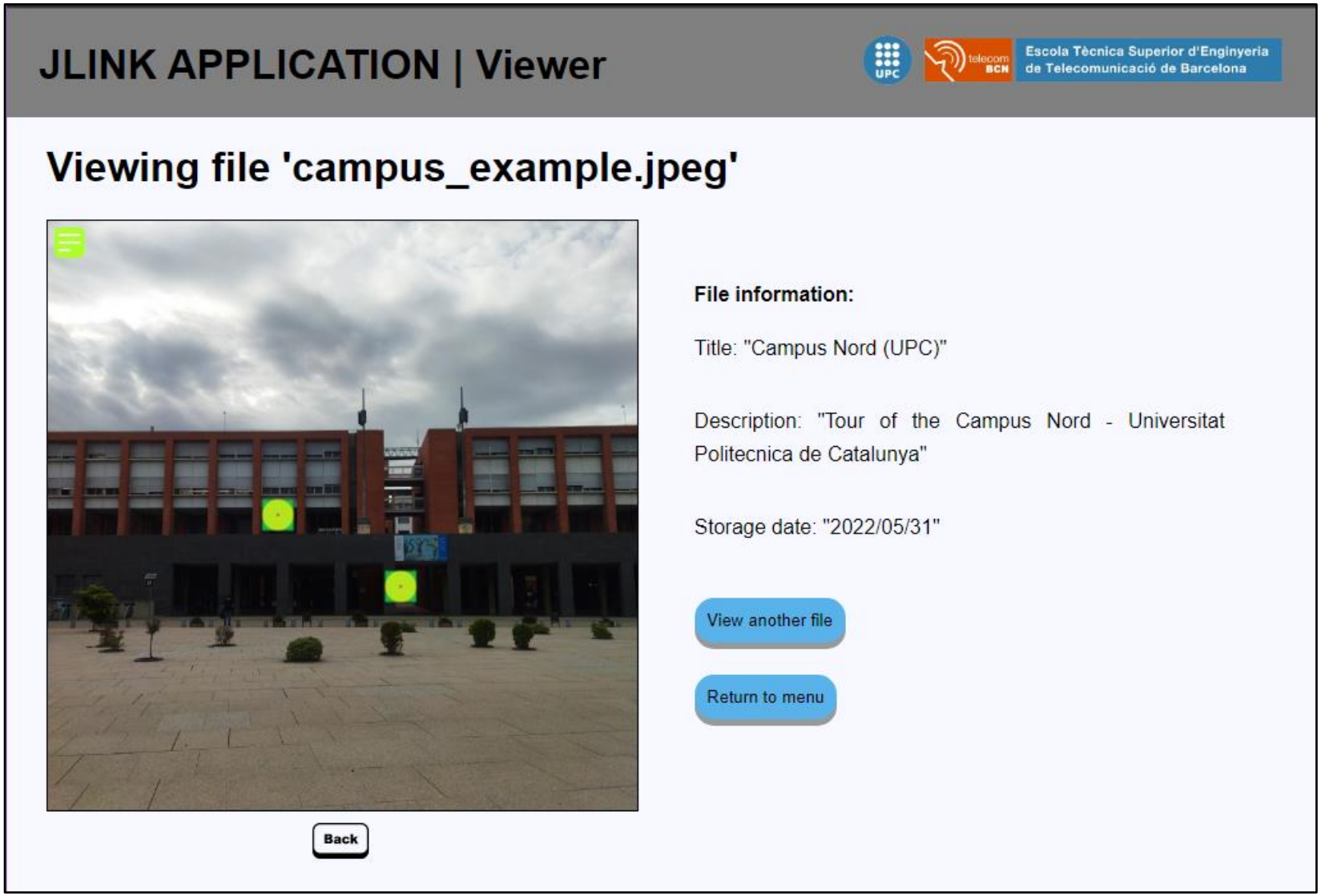


Figure D5. Image 5 from Figure 9.

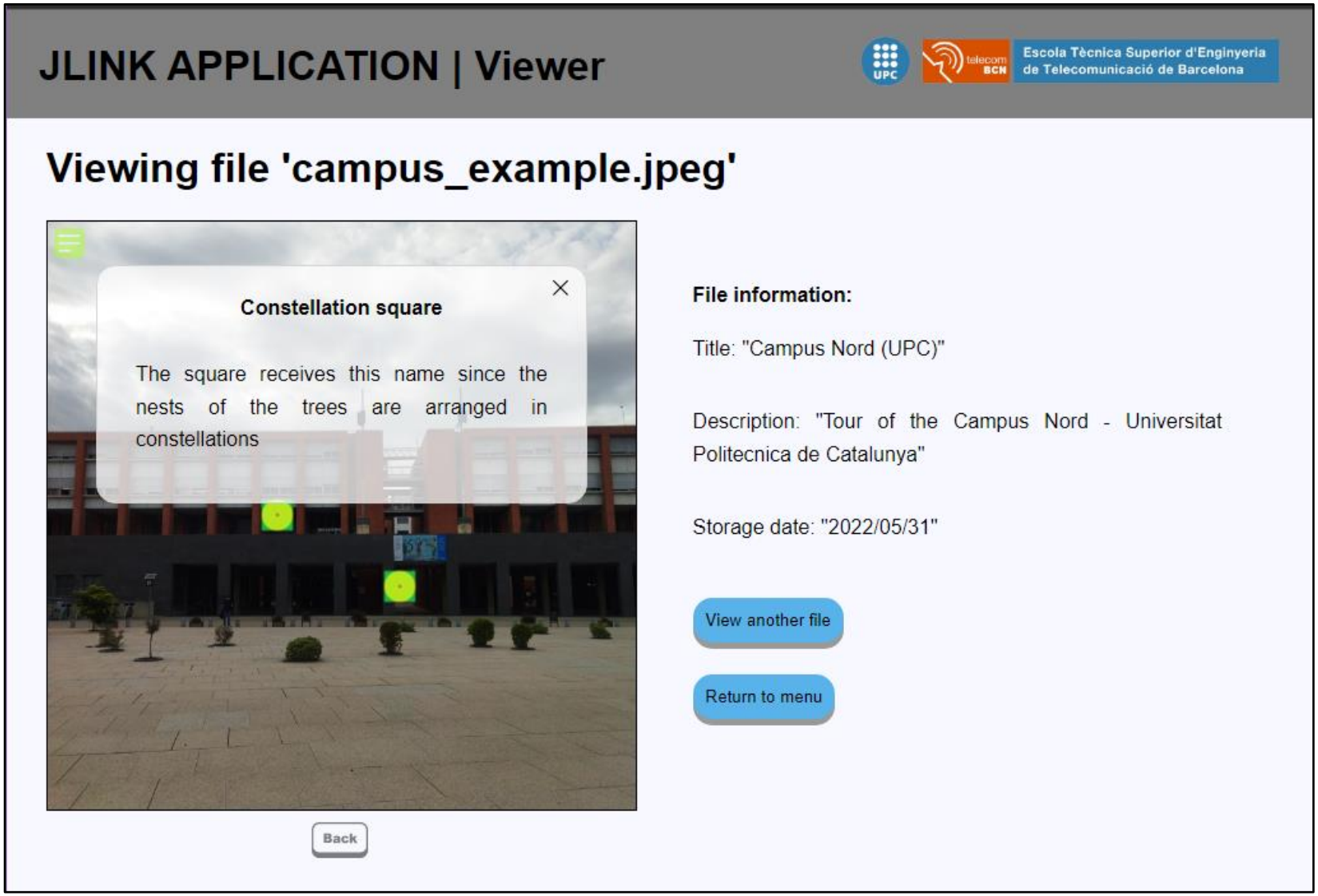


Figure D6. Image 6 from Figure 9.

Appendix E: JLINK Modifier Images

This appendix shows in detail the screenshots of Figure 10.

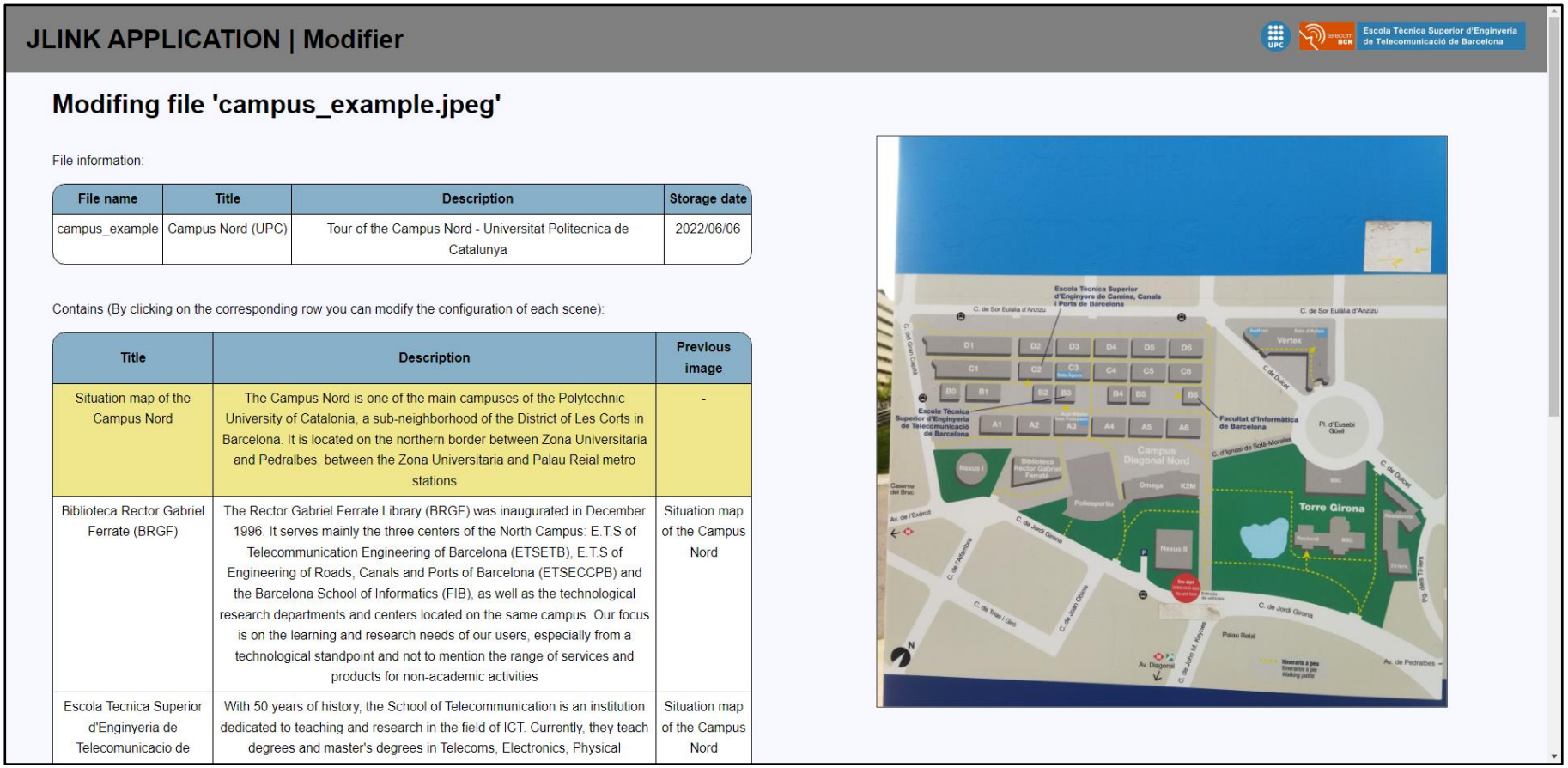


Figure E1. Image 1 from Figure 10.

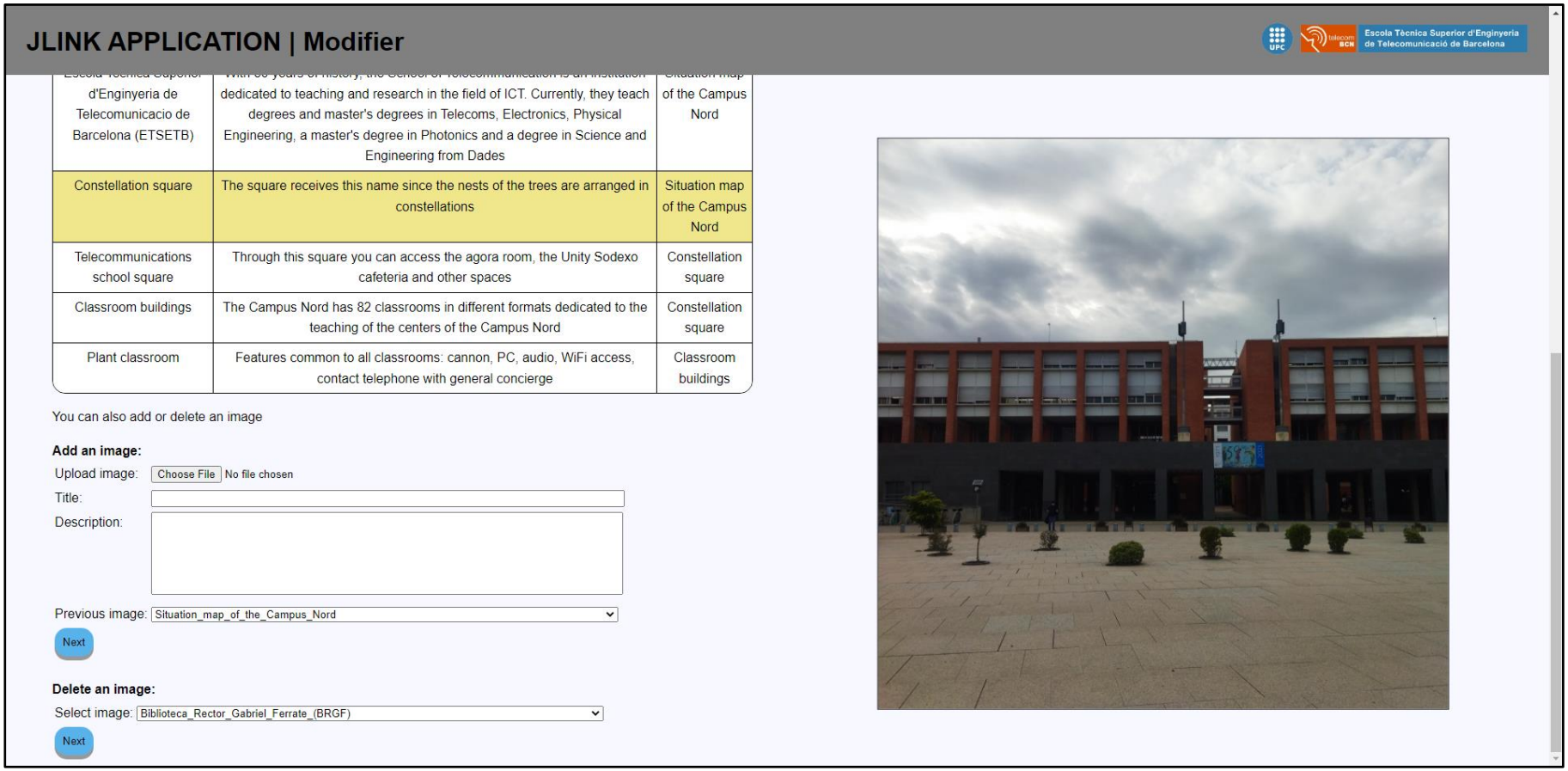


Figure E2. Image 2 from Figure 10.

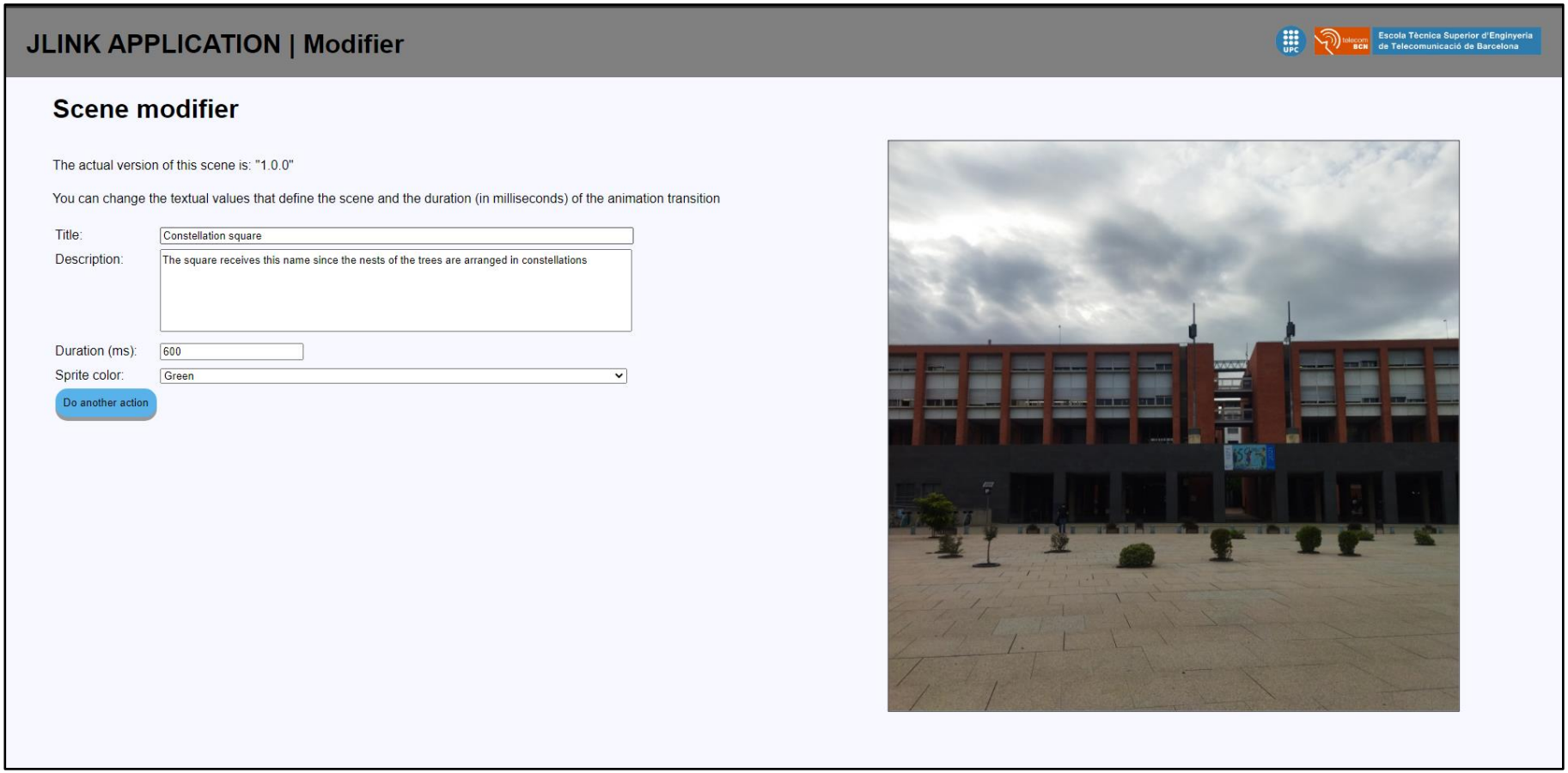


Figure E3. Image 3 from Figure 10.

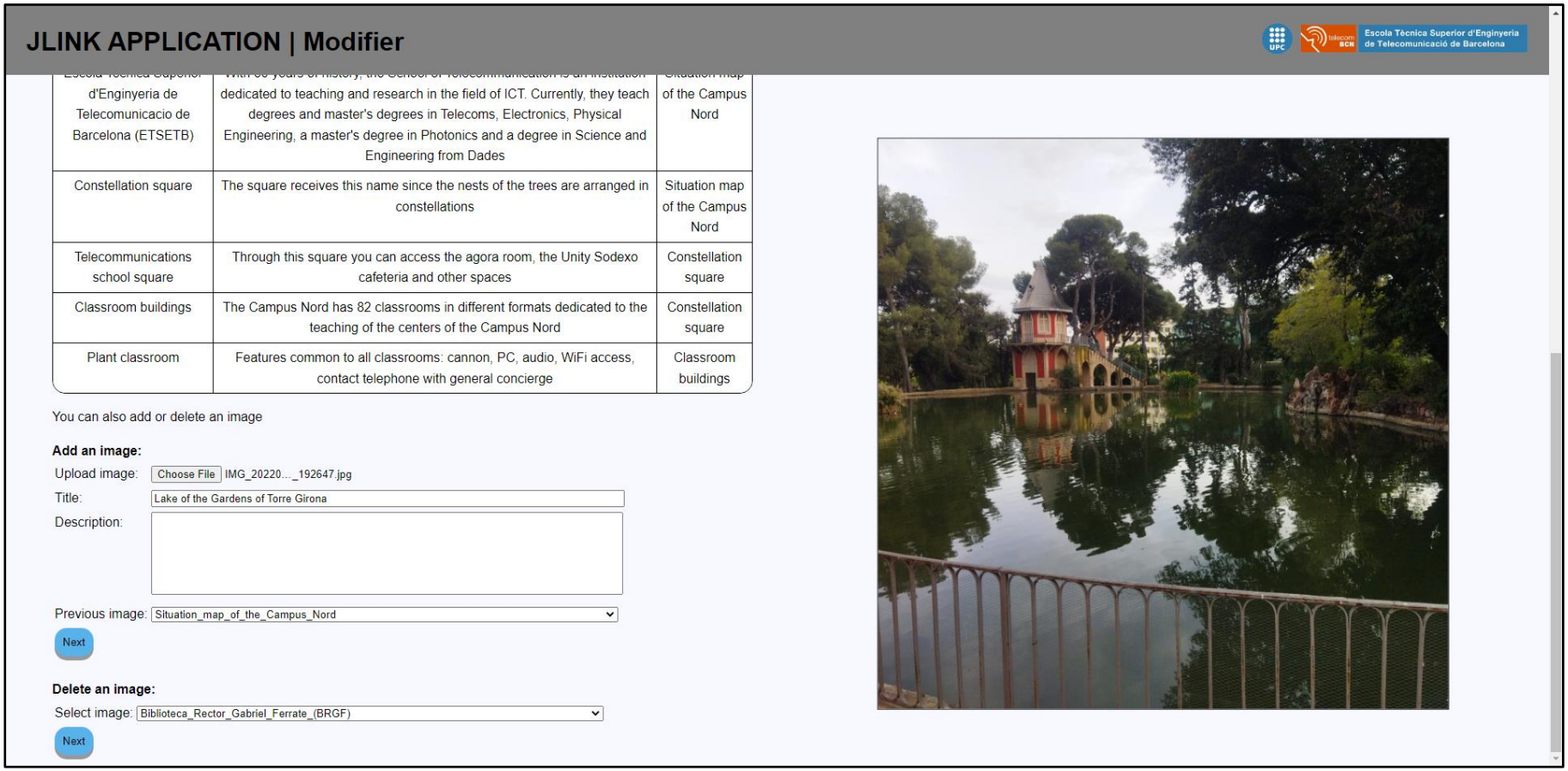


Figure E4. Image 4 from Figure 10.

JLINK APPLICATION | Modifier

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Creating file 'campus_example.jpeg'

Image size:

- Width: "3000 px"
- Height: "3000 px"

Select sprite:

- Coord X: "2039 px"
- Coord Y: "2133 px"

Do another action

Figure E5. Image 5 from Figure 10.

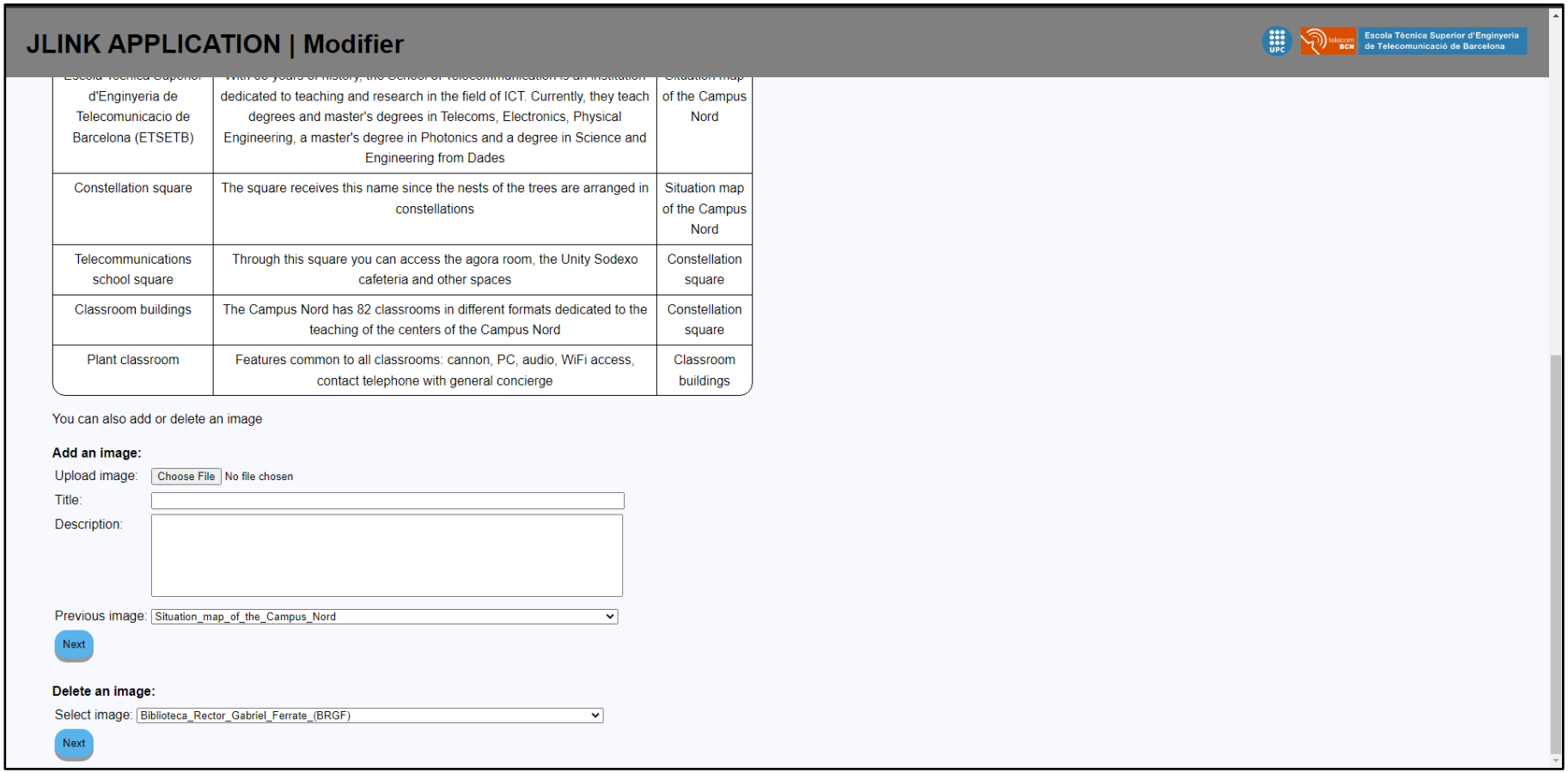


Figure E6. Image 6 from Figure 10.

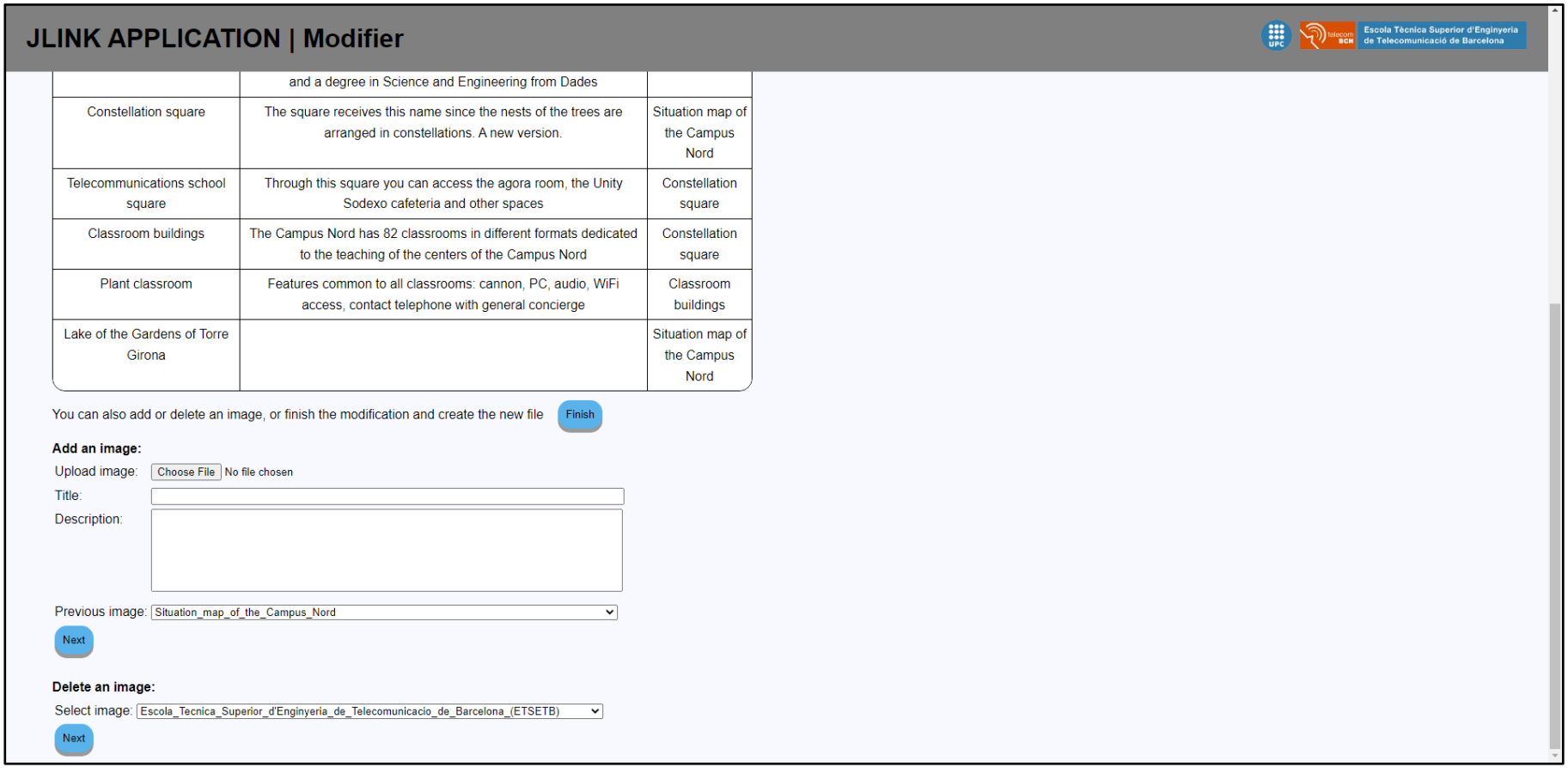


Figure E7. Image 7 from Figure 10.

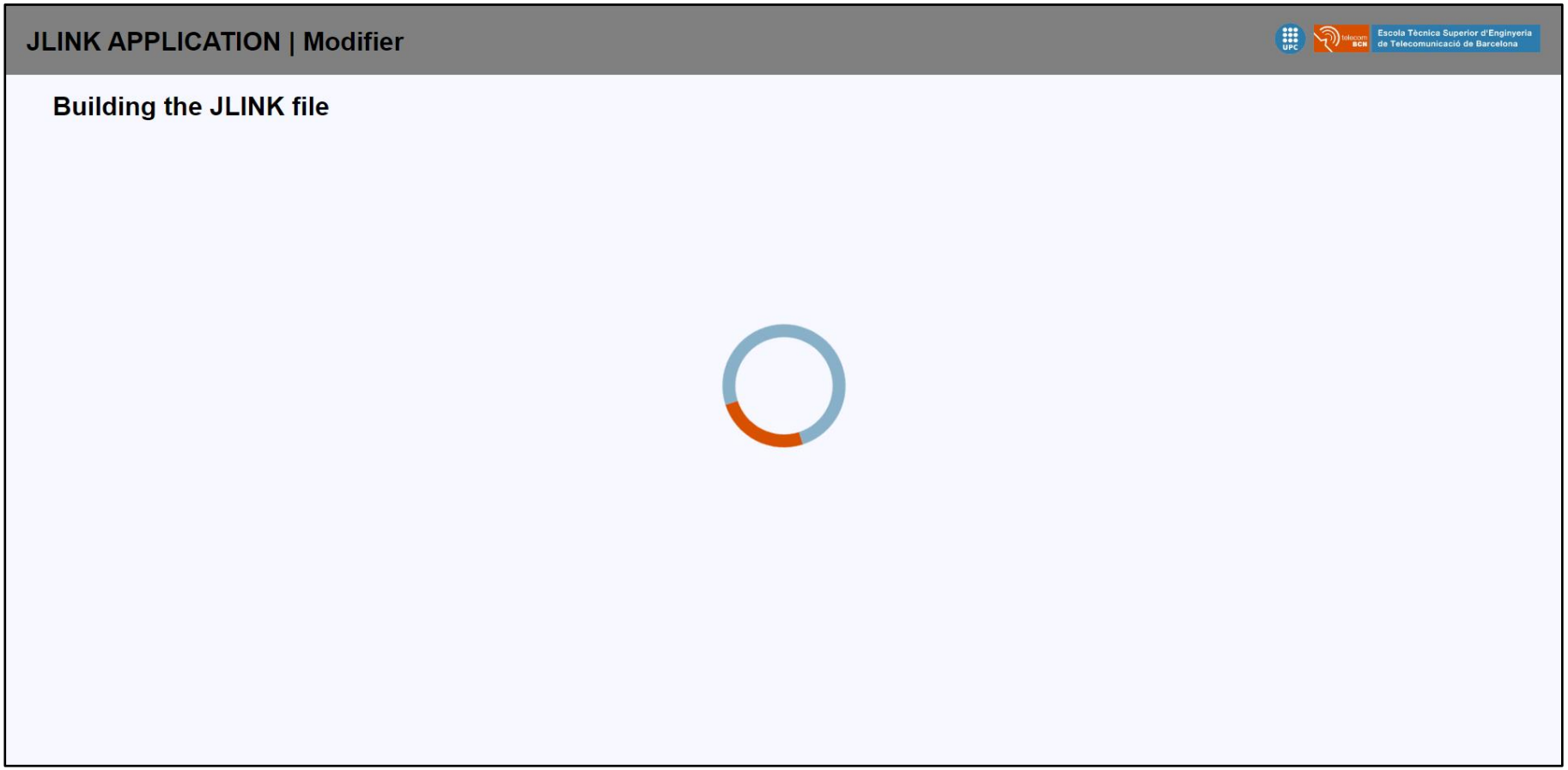


Figure E8. Image 8 from Figure 10.

JLINK APPLICATION | Modifier

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

JLINK Modifier

The file has been created successfully.

File information:

File name	Title	Description	Storage date
campus_example_v1	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/06/15

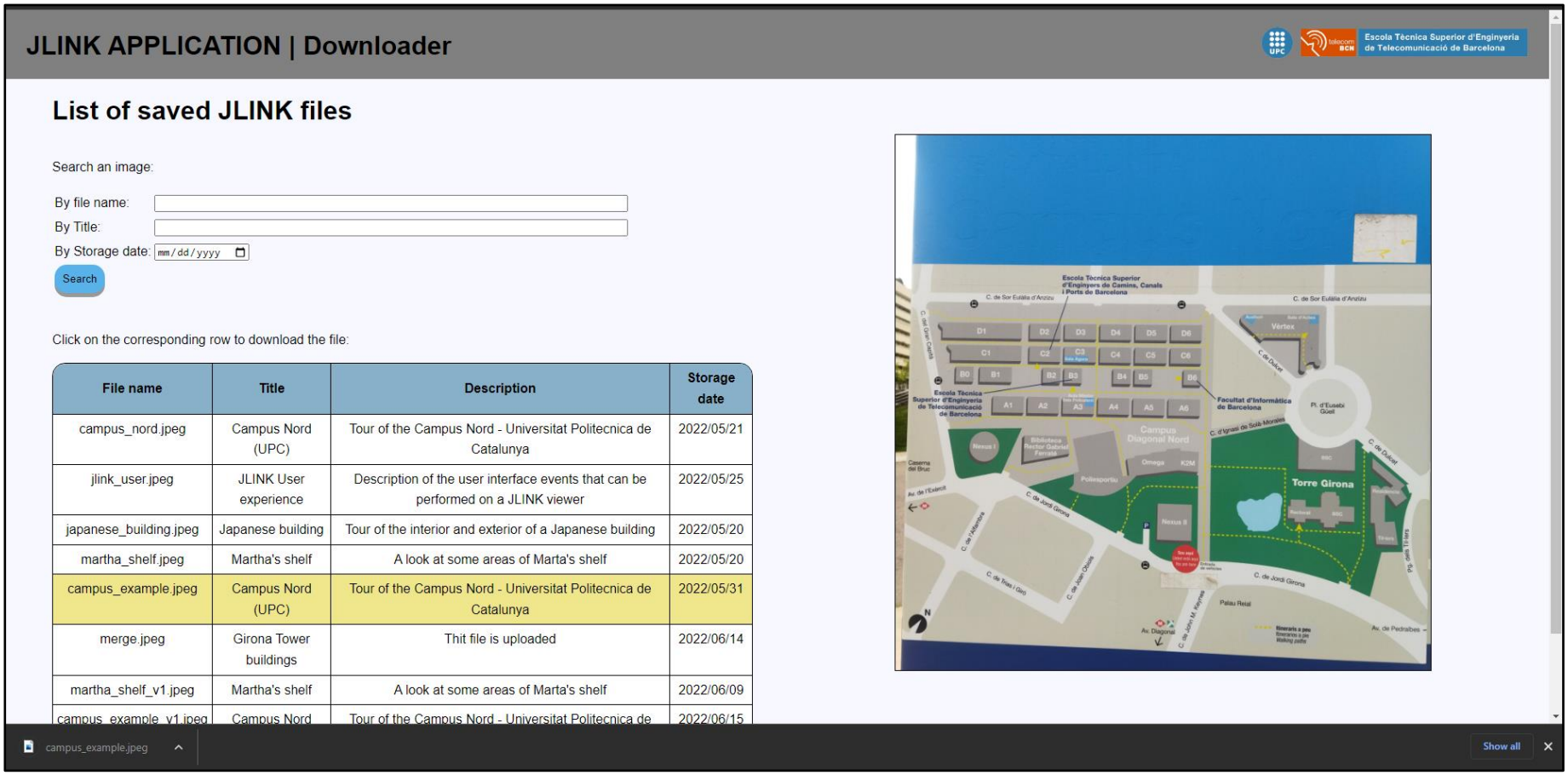
Contains:

Title	Description	Previous image
Situation map of the Campus Nord	The Campus Nord is one of the main campuses of the Polytechnic University of Catalonia, a sub-neighborhood of the District of Les Corts in Barcelona. It is located on the northern border between Zona Universitaria and Pedralbes, between the Zona Universitaria and Palau Reial metro stations	-
Escola Tecnica Superior d'Enginyeria de Telecomunicacio de Barcelona (ETSETB)	With 50 years of history, the School of Telecommunication is an institution dedicated to teaching and research in the field of ICT. Currently, they teach degrees and master's degrees in Telecoms, Electronics, Physical Engineering, a master's degree in Photonics and a degree in Science and Engineering from Dades	Situation map of the Campus Nord
Constellation square	The square receives this name since the nests of the trees are arranged in constellations	Situation map of the Campus Nord
Telecommunications school square	Through this square you can access the agora room, the Unity Sodexo cafeteria and other spaces. A new version.	Constellation square

Figure E9. Image 9 from Figure 10.

Appendix F: JLINK Downloader Image

This appendix shows in detail the screenshot of Figure 11.

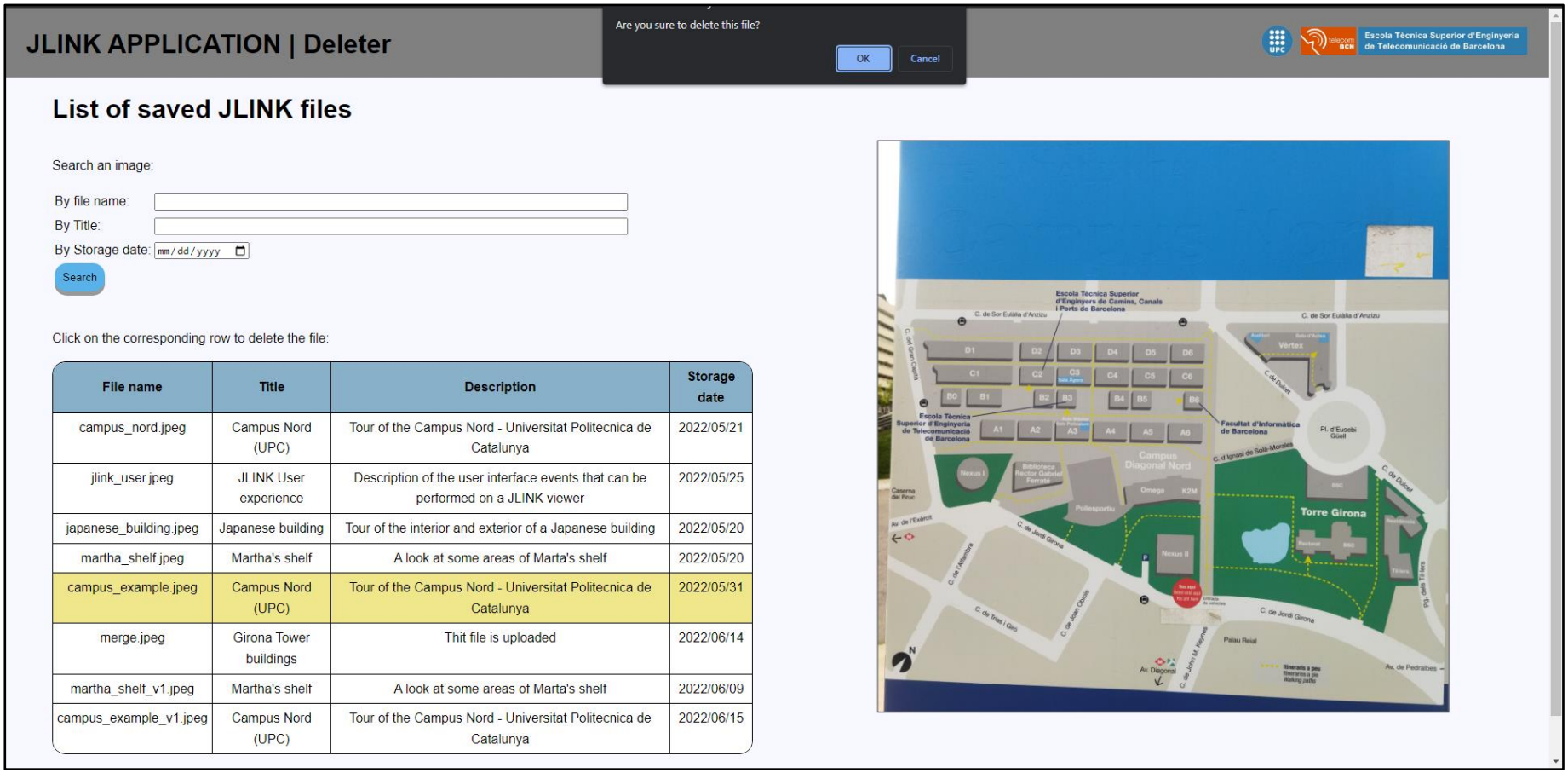


File name	Title	Description	Storage date
campus_nord.jpeg	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/05/21
jlink_user.jpeg	JLINK User experience	Description of the user interface events that can be performed on a JLINK viewer	2022/05/25
japanese_building.jpeg	Japanese building	Tour of the interior and exterior of a Japanese building	2022/05/20
martha_shelf.jpeg	Martha's shelf	A look at some areas of Marta's shelf	2022/05/20
campus_example.jpeg	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/05/31
merge.jpeg	Girona Tower buildings	Thit file is uploaded	2022/06/14
martha_shelf_v1.jpeg	Martha's shelf	A look at some areas of Marta's shelf	2022/06/09
campus_example_v1.jpeg	Campus Nord	Tour of the Campus Nord - Universitat Politecnica de	2022/06/15

Figure F1. Image from Figure 11.

Appendix G: JLINK Deleter Image

This appendix shows in detail the screenshot of Figure 12.



File name	Title	Description	Storage date
campus_nord.jpeg	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/05/21
jlink_user.jpeg	JLINK User experience	Description of the user interface events that can be performed on a JLINK viewer	2022/05/25
japanese_building.jpeg	Japanese building	Tour of the interior and exterior of a Japanese building	2022/05/20
martha_shelf.jpeg	Martha's shelf	A look at some areas of Marta's shelf	2022/05/20
campus_example.jpeg	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/05/31
merge.jpeg	Girona Tower buildings	Thit file is uploaded	2022/06/14
martha_shelf_v1.jpeg	Martha's shelf	A look at some areas of Marta's shelf	2022/06/09
campus_example_v1.jpeg	Campus Nord (UPC)	Tour of the Campus Nord - Universitat Politecnica de Catalunya	2022/06/15

Figure G1. Image from Figure 12.

.

Glossary

Box: binary structure that encapsulates an object embedded in a file.

Codestream: sequence of bits representing a compressed image and associated metadata.

JPEG: Joint Photographic Experts Group.

JUMBF: JPEG Universal Metadata Box Format.

JLINK: JPEG Linked Media Format.

Metadata: data about data.

Link: relational description of a scene to another scene composed of linkage region, sprite, and visual effect for scene change.

Scene: basic unit of a JLINK composed of a 2D image and text.

Sprite: visual element on the user display which indicates user interaction events.

Superbox: box that only contains other boxes.

XML: Extensible Markup Language.