Grant Agreement n.: LC-01409964

FEAT

Project full title: “Features for Accessibility through Templates”

Call identifier: PAWA-2019

Topic: Preparatory Action: Application of web accessibility requirements in web-authoring tools and platforms by default

D1.2 Specifications for the Episerver templates, including a list of selected accessibility features

Deliverable Id:	D1.2
Deliverable Name:	Specifications for the Episerver templates, including a list of selected accessibility features
Version:	1.0
Status:	FINAL
Due date of deliverable:	30 June 2020
Actual submission date:	30 June 2020
Work Package:	1. User requirements and specifications
Organisation name of lead partner for this deliverable:	Funka
Author(s):	Susanna Laurin, Anders Beinhoff, Jakob Hasslöw
Partner(s) contributing:	N.A

Abstract: The Specifications for the Episerver templates, including a list of selected accessibility features, describes the templates to be developed in prototype form for testing. Previously gathered user requirements are analysed and prioritised from the perspective of technical feasibility.

Disclaimer:

FEAT has received funding from the European Commission’s Work Programme for 2019 for Pilot Projects and Preparatory Actions in the field of Communications Networks, Content and Technology under grant agreement No LC-01409964

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Commission. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

This deliverable contains original unpublished work or work to which the author holds all rights except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of contents

List of figures

List of table

Executive summary

This report summarises the specifications for the Episerver templates, including a list of selected accessibility features in the project FEAT.

The purpose of the activity is to analyse the general, as well as Episerver specific, user requirements from D1.1. This deliverable will explore how these requirements can be best translated into templates with accessibility features that are ready to use for Episerver partners and clients, as well as providing inspiration for other authoring tools.

The aim has been to select templates that cover:

- the most common accessibility fails.
- the objects that Episerver web authors struggle the most with.
- the items where fails can harm the end users most.
- the functions that are technically feasible.

Based on input from:

- web authors using Episerver.
- the company Episerver who are developing the core of the tool.
- Episerver partners who are using the tool to develop websites.
- the experience from the research team during 20 years of accessibility testing and auditing and 15 years as an Episerver partner.

A set of accessibility by default features have been selected for prototyping and testing.

The selection resulted in a set of proposed templates that cover a mix of pages, objects and blocks, where the testing will be focused on how to best meet the requirements of the web authors.

Next steps include prototyping the selected features and development of user tests scenarios. The research team will also start booking participants for the testing phase among Episerver web authors in the public sector.

1. Introduction

1.1. Scope and objective of deliverable

This deliverable describes the work done in analysing the user requirements of D1.1. The user requirements that have been gathered will be sorted, prioritised and analysed from the perspective of technical feasibility. Technical feasibility will be assessed including both factors related to the Episerver framework and broader factors that are common to the core functionalities of different authoring tools. The result will be a list of accessibility features to be developed as prototypes ready for testing.

The specifications will pay special attention to identifying which accessibility features can be implemented as automated features and which features can be made accessible by providing guidance to the web author in a semi-automated way. The objective is to automate as many accessibility features as possible in order to minimise the risk of errors and leave room for web authors to concentrate on content production.

1.2.Methodology of work

In this action, data and information gathered in D 1.1, essentially user needs and requirements from Episerver web authors and Episerver partners, are sorted, prioritised and analysed from the perspective of technical feasibility. Each user need that can be converted into an accessibility feature is assessed for technical feasibility factors related to the Episerver framework, as well as broader factors that are common to the core functionalities of different authoring tools, based on the results of the WE4AUTHORS pilot.

The selection criteria used for this process are based on how Episerver is used and how the authoring relates to accessibility, from different stakeholder perspectives:

- Most commonly used templates, pages and objects.
- Functionality where accessibility problems are most often seen according to partners.
- Functionality where accessibility issues are causing most barriers according to end users.
- Functionality where accessibility is perceived as difficult according to web authors.

The objective is to select a set of templates that provide a good starting point to develop a basic information website in the public sector.

Special attention is also paid to identifying which accessibility features can be implemented as automated features and which features can be made accessible by providing guidance to the web author in a semi-automated way. Automation is more efficient to minimise the risk of errors and leave room for web authors to concentrate on content production. It is also preferred by web authors. However, it is not always possible to achieve full automation at the same time as the website aims to be flexible and dynamic. Therefore, some accessibility features will be supporting the author in doing the right thing.

2. Episerver structure

The platform for Episerver websites is ASP.NET MVC, a framework for web applications developed by Microsoft. The platform is open and the MSFT .NET technology enables integrations to other systems. The Application Programming Interfaces (APIs) are abundant, which makes it easy for developers to connect systems to each other quickly and smoothly.

To develop a website on Episerver, the website owner needs an Episerver partner to create the Episerver website. The Episerver partner has a set of base classes to choose from when starting a new project. The base classes include a minimum basic functionality, but are otherwise empty. This provides the Episerver partner with great freedom and flexibility to develop the templates, pages and functionality. The Episerver code library (Episerver.Core) provides a hierarchy of functionality and objects, that can be inherited throughout the structure of the website.

The selected templates will be developed based on Episerver base classes, re-using MSFT .NET technology.When it comes to fulfilling responsibility and requirements for accessibility, the above structure is key. The Episerver partner doesn't have to handle any inaccessible code that comes with the authoring tool and no automatic updates of the tool are made. This is important because automatic updates risk overriding existing accessibility remediations made in the code. Many authoring tools have this problem. However, in the case of Episerver, the partner decides how and when to make the updates, which means they have full control over the process.

The authoring interface of Episerver gives the author a good overview of the information structure of the entire website. The web author can choose to publish and update content using the drag & drop function or by writing content manually.

3. Web author needs

Based on results from the stakeholder workshops carried out in the pilot project, the knowledge gap, in combination with difficulties/time needed to perform the steps to achieve accessibility, are recurring responses when deciding which part of the authoring process to prioritise.

To further deepen the understanding of specific needs for Episerver authors, the online survey focused on Episerver-specific questions, to make sure the templates prototyped and tested were relevant to web authors.

3.1.Online survey

In the D1.1 report on user needs, web authors in public sector using Episerver were asked about the most difficult areas of publishing in an accessible way. The following items were listed (multiple choice question):

Features	% of respondents
Documents	78
Tables	60
Images	48
Forms	46
Videos	42
Links	40

Navigation	32
Text	32
Headings	32

Table 1: Publishing difficulties answers from survey

The top two choices, documents and tables, are not in the scope of this project, as none of them are handled by the Episerver templates.

When asked to point out the specific parts of Episerver that could be beneficial to have as built-in accessibility features, most respondents claimed “all parts” should have built-in accessibility support. Among the respondents who chose to specify objects, the following items were listed:

Features	% of respondents
Article page	45
Text block	12
Epi forms	10
Start page	6
Videos	4
Document listing	2

Table 2: Survey answers regarding built-in accessibility support

As the majority of respondents claimed the article page to be the most important, several objects and blocks that make up the article page (such as image handling, calendar events etc.) are also covered by the survey results.

3.2. Automation vs support

According to the workshop results in the pilot project, clear instructions, manuals and support are highlighted by both web authors with some knowledge on accessibility and those who are new to the subject. Ease of use, intuitive interfaces and built-in accessibility by default are generally seen as more import by website owners and the central communication offices who have to remediate the same mistakes over and over again. Experienced authors and website owners highlight that a combination of semi-automatic features with reminders and instructions may be the most feasible option, as full automation and mandatory fields tend to be less flexible in everyday use.

Looking at the prioritised list of items by Episerver web authors in the survey, some preliminary analysis has been done on how to maximise the support for authors:

- The basic accessibility of all objects in an article page or start page can be built-in by default.
- Items such as listings can be fully automated.

- Videos can provide a way to add captions, but information and support on how to use third party automatic captioning functionality in the most efficient way – or indeed how to caption manually - needs instructions.
- ALT-text fields is an example of items that would be semi-automated. For example, the field is made to be mandatory and appears automatically. If the author doesn't fill in the ALT-text, the image doesn't show. This is of course provides strong support for a wider use of ALT-texts. However, it could also result in web authors rejecting to use images at all, if they find the handling too cumbersome.

The level of automation, the extent to which it is desirable to force authors to comply and the granularity of support/information are examples of perspectives that the user testing of prototypes will focus on.

4. Technical feasibility

In order to make sure that the result of the project will lead generate maximum impact; the objective is to select a set of templates that provides a good starting point to develop a basic information website in the public sector. This means covering page types and functionality that are usually found in these kinds of websites.

By using block functions, it will be possible to re-use the accessibility features in different page templates, providing flexibility to the web authors and solid support when creating an accessible website. By dividing the objects into fixed, accessible characteristics, accessible content creation is simplified. This process makes it easier for the web author to ensure that concept, design patterns and accessibility levels are kept intact no matter when the default accessibility features are used.

This will not only help the web authors in their daily work, but it also means that the technical accessibility knowledge of the supplier becomes less important. The division into smaller parts is also helpful for the wider uptake of the templates in other authoring tools, paving the way for inspiration and innovation in all tools built on templates.

5. Selection criteria

Based on the pilot project results, the findings in D1.1 and the experience of the research team, a set of criteria has been developed to ensure the market uptake of the templates developed in the project. The criteria are listed below:

- Frequency of use – the templates, pages and objects should be based on items that web authors in the public sector often use for creating content.
- Frequency of accessibility fails – Funka has done thousands of accessibility audits over the years and has a clear view of which problems occur most often, including specific expertise on Episerver websites.
- Technical feasibility – the set-up of the templates should build upon a process for content creation that provides flexibility for the author, resulting in consistency and stability of the website.
- Perceived difficulty in making it accessible – based on workshop and survey results, the templates should address the items that authors prioritise.

- Potential impact on end users with disabilities – the templates should try to cover (and make the author avoid) the most important potential accessibility fails to end users. Items that create the worst barriers should be included in the templates.

6. Selection of featured and templates

Based on the selection criteria, the following features have been selected for prototyping and testing.

- Framework
- Start page
- Article page
- Calendar event page
- Page listing block
- Calendar listing block
- Text block
- Image block
- Forms block
- Video block

The features are described more in detail below.

6.1.Framework template

The framework of the website consists of a number of common and central modules which in most cases re-appear in all pages of the website. The logo, the navigation and the footer are examples of typical framework items. As the framework items are reachable from all parts of the website, it is key to make sure the framework is accessible.

From an accessibility perspective, it is valuable to use a modular design and set up of the different parts of the framework for two reasons:

1) the end user will quickly learn the concept on the particular interface and can soon use the functions of the framework intuitively.
2) The web author only has to create the content once, which minimise the risk of mistakes.

Making it easy to do the right thing increases the chance for web authors to succeed in publishing accessible content. The accessible features of the framework template selected for prototyping and testing include the following functions:

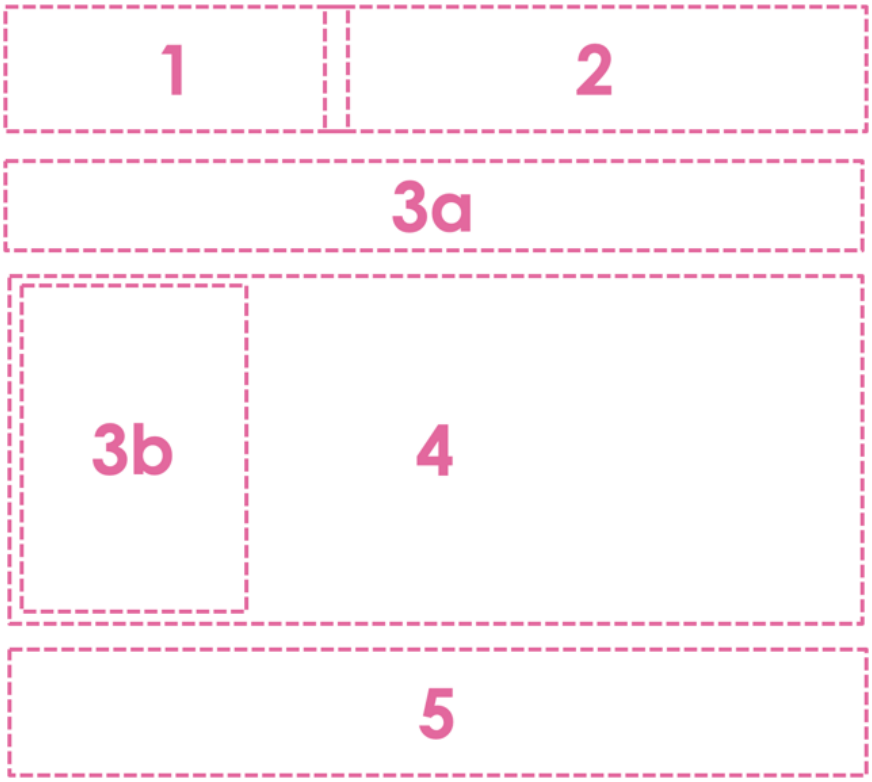


Figure 1: Example of framework templates in a wireframe

1 & 2 Page header

The website page header is designed so that the end users, no matter where they are in the website structure can easily reach the central functions.

Logotype

The logo of the website owner is placed by using the “add image” function of the page header. This gives the website a clear sender, which is important for the end users to be able to decide if this is the right website or not. The “add image” function includes a field for ALT-text. The logo is also linked to the start page of the website, as a standard practice.

Short cuts

The possibility to create accessible short cuts is important for web authors to be able to offer functions to activate built-in assistive technology or to change the language of the website.

Quick search

As user preferences vary when it comes to navigating the website via content navigation or search functionality, it is best practice to place the search function close to the navigation. The accessible search field is automatically created and doesn’t need to be altered by the author.

3a Navigation concept

In desktop mode, the accessible navigation concept is shown as a set of parallel main entry points with sub-sections hidden until clicked. When the end user choses an entry point, the selected page is opened, and the sub sections of the navigation menu are shown to the left. This navigation concept is created for maximum accessibility and has been tested with large numbers of end users with different disabilities.

In the responsive mode, the accessible navigation concept is shown as a collapsable so-called hamburger menu. Once clicked, the sub-sections are shown in a separate list of the interface.

The accessible navigation concept is automatically indexed covering the complete content structure of the website. As both the number and the naming of the main entry points are important for end users, different set ups will be prototyped and tested. The objective is to see which approach is better to achieve a limited number of entry points and names that are not too long.

3b & 4 Page content

Area 4 in the illustration consists of content that is not part of the framework. However, one section in this area does form part of the framework; the sub-sections of the navigation concept in 3b. This is shown in all level 2 pages of the website. In this area, sub-sections of the navigation are shown as a list to the left of the screen. The prototyping and testing will show whether it is a good idea to have this activated by default or not.

The accessible listing automatically indexes the pages of the website hierarchy in a structured way. The author doesn’t need to do any manual work.

This feature ensures consistency of the website navigation which is an important part of accessibility.

The area containing each specific page type (4) is described in the following chapters.

5 Footer

The accessible footer of the framework is the lower ending point of all pages of the website. The footer is used for information and links that are not part of the information structure, but that are important to be able to reach globally, although not deemed as high priority. Typical content of a footer includes contact information, subscription functionality for newsletters and links to social media channels.

The footer is going to be prototyped and tested with a number of characteristics including the possibility to edit text and add links.

6.1.1. List of WCAG success criteria covered in the framework template

Specific WCAG criteria for the framework template
Success Criterion 1.3.1 Info and Relationships (Level A)
Success Criterion 1.3.2 Meaningful Sequence (Level A)
Success Criterion 1.3.3 Sensory Characteristics (Level A)
Success Criterion 1.3.4 Orientation (Level AA)
Success Criterion 1.3.5 Identify Input Purpose (Level AA)
Success Criterion 1.4.4 Resize text (Level AA)
Success Criterion 1.4.10 Reflow (Level AA)
Success Criterion 1.4.12 Text Spacing (Level AA)
Success Criterion 1.4.13 Content on Hover or Focus (Level AA)

Success Criterion 2.1.1 Keyboard (Level A)
Success Criterion 2.1.2 No Keyboard Trap (Level A)
Success Criterion 2.1.4 Character Key Shortcuts (Level A)
Success Criterion 2.4.1 Bypass Blocks (Level A)
Success Criterion 2.4.2 Page Titled (Level A)
Success Criterion 2.4.3 Focus Order (Level A)
Success Criterion 2.4.5 Multiple Ways (Level AA)
Success Criterion 2.4.6 Headings and Labels (Level AA)
Success Criterion 2.4.7 Focus Visible (Level AA)
Success Criterion 2.5.2 Pointer Cancellation (Level A)
Success Criterion 2.5.3 Label in Name (Level A)
Success Criterion 3.1.1 Language of Page (Level A)
Success Criterion 3.2.1 On Focus (Level A)
Success Criterion 3.2.2 On Input (Level A)
Success Criterion 3.2.3 Consistent Navigation (Level AA)
Success Criterion 3.2.4 Consistent Identification (Level AA)
Success Criterion 4.1.1 Parsing (Level A)
Success Criterion 4.1.2 Name, Role, Value (Level A)
Success Criterion 4.1.3 Status Messages (Level AA)

WCAG criteria that can be affected by authors using the framework template
Success Criterion 1.1.1 Non-text Content (Level A)
Success Criterion 1.4.5 Images of Text (Level AA)
Success Criterion 2.4.4 Link Purpose (In Context) (Level A)
Success Criterion 3.1.2 Language of Parts (Level AA)

6.2. Start page

The start page is for many end users the first page they will find, as the URL will take them here. How the start page succeeds in combining accessibility, concept, presentation, content and ”feeling” is key to ensuring that the end users stay around. Even if the technical set up is accessible, end users also need the other components to be harmonised.

Due to the importance of the start page from the end user perspective, and because the start pages differ so much among organisations, the features and areas must be flexible and easily adaptable. In the content areas, or so-called block areas, the web author can easily create blocks of functions or teasers.

The start page contains the possibility to push for specific content via different kinds of teasers. These are created by accessible functionality blocks, which are dynamic when it comes to layout, controlled by size and placement. For example, the author can choose to have 2 large teasers in a row or 3 smaller ones, with or without images.

The start page is still important for many users, even though the general user pattern has changed. These days, most end users comes directly to a specific page on the website, directed there by a search engine – without passing through the start page.

The accessible features of the start page template selected for prototyping and testing include the following functions:

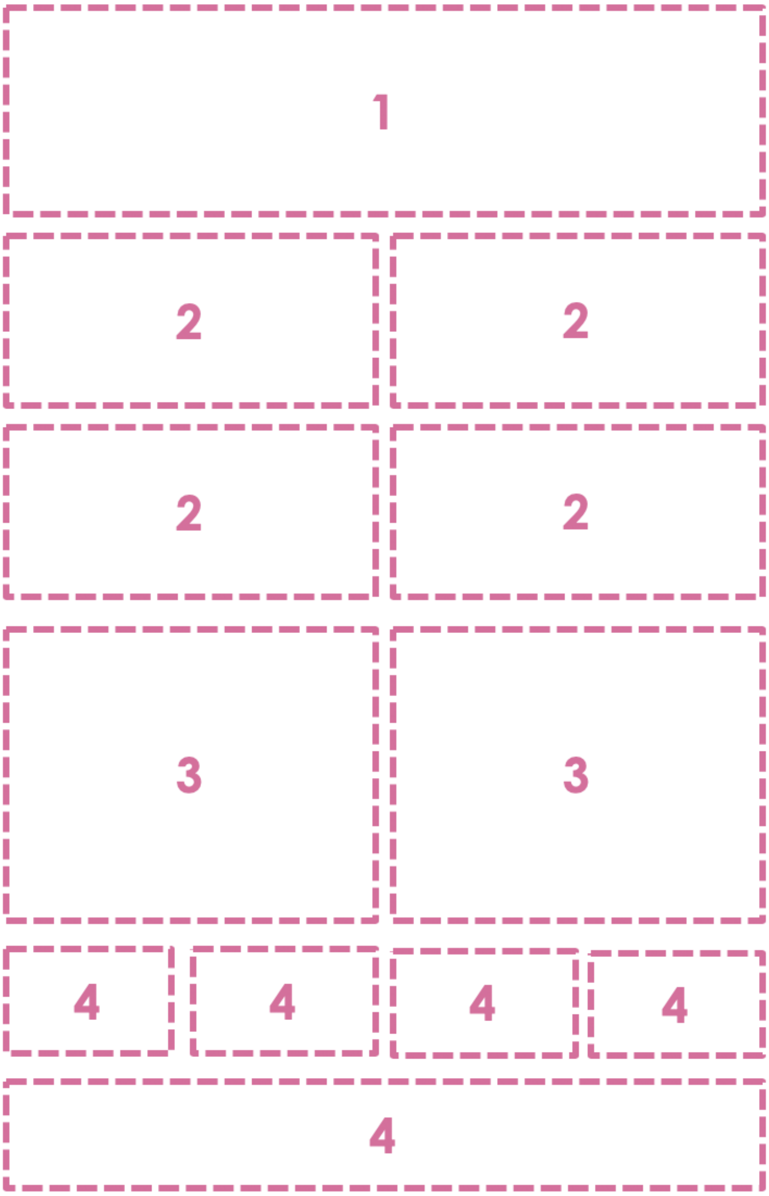


Figure 2: Example of wireframe for start page

Top section (1)

The user needs a quick and clear confirmation that he or she has ended up in the right place, the top section is a natural area to welcome the user with text and images to make it clear who is the sender/website owner. This should be the first thing the sighted user should see on an average screen and/or resolution, without scrolling.

Middle sections (2-4)

The main objective of the start page is to give an overview of the content and push for specific items. In the middle section, the web author can choose to highlight key content or news with different kinds of accessible teasers or listings. The blocks are flexible and can be used in different combinations of sizes, depending on content.

6.2.1. List of WCAG success criteria covered in the start page template

Specific WCAG criteria for the start page template
Success Criterion 1.3.1 Info and Relationships (Level A)
Success Criterion 1.3.2 Meaningful Sequence (Level A)
Success Criterion 1.3.3 Sensory Characteristics (Level A)
Success Criterion 1.3.4 Orientation (Level AA)
Success Criterion 1.3.5 Identify Input Purpose (Level AA)
Success Criterion 1.4.4 Resize text (Level AA)
Success Criterion 1.4.10 Reflow (Level AA)
Success Criterion 1.4.12 Text Spacing (Level AA)
Success Criterion 1.4.13 Content on Hover or Focus (Level AA)
Success Criterion 2.1.1 Keyboard (Level A)
Success Criterion 2.1.2 No Keyboard Trap (Level A)
Success Criterion 2.1.4 Character Key Shortcuts (Level A)
Success Criterion 2.4.1 Bypass Blocks (Level A)
Success Criterion 2.4.2 Page Titled (Level A)
Success Criterion 2.4.3 Focus Order (Level A)
Success Criterion 2.4.5 Multiple Ways (Level AA)
Success Criterion 2.4.6 Headings and Labels (Level AA)
Success Criterion 2.4.7 Focus Visible (Level AA)
Success Criterion 2.5.2 Pointer Cancellation (Level A)
Success Criterion 2.5.3 Label in Name (Level A)
Success Criterion 3.1.1 Language of Page (Level A)
Success Criterion 3.2.1 On Focus (Level A)
Success Criterion 3.2.2 On Input (Level A)
Success Criterion 3.2.3 Consistent Navigation (Level AA)
Success Criterion 3.2.4 Consistent Identification (Level AA)
Success Criterion 4.1.1 Parsing (Level A)
Success Criterion 4.1.2 Name, Role, Value (Level A)
Success Criterion 4.1.3 Status Messages (Level AA)

General WCAG criteria that can be affected by authors using the start page template

Success Criterion 1.1.1 Non-text Content (Level A)
Success Criterion 1.4.5 Images of Text (Level AA)
Success Criterion 2.4.4 Link Purpose (In Context) (Level A)
Success Criterion 3.1.2 Language of Parts (Level AA)

6.3. Article page

The article page is the most frequently used page, where most of the content of the website will be presented. It will be designed as an accessible template with a set of basic and accessible features. To provide the author with maximum flexibility, the idea is to combine fixed parts with block areas where the content can be created directly. The navigation concept will be fully implemented in the prototype article page.

The prototype will use default patterns in the article page to facilitate design for the web author, making content creation easier. At the same time as the end user will benefit from consistency, which makes it easier to learn, understand and use the interface.

The accessible features of the article page template selected for prototyping and testing include the following functions:

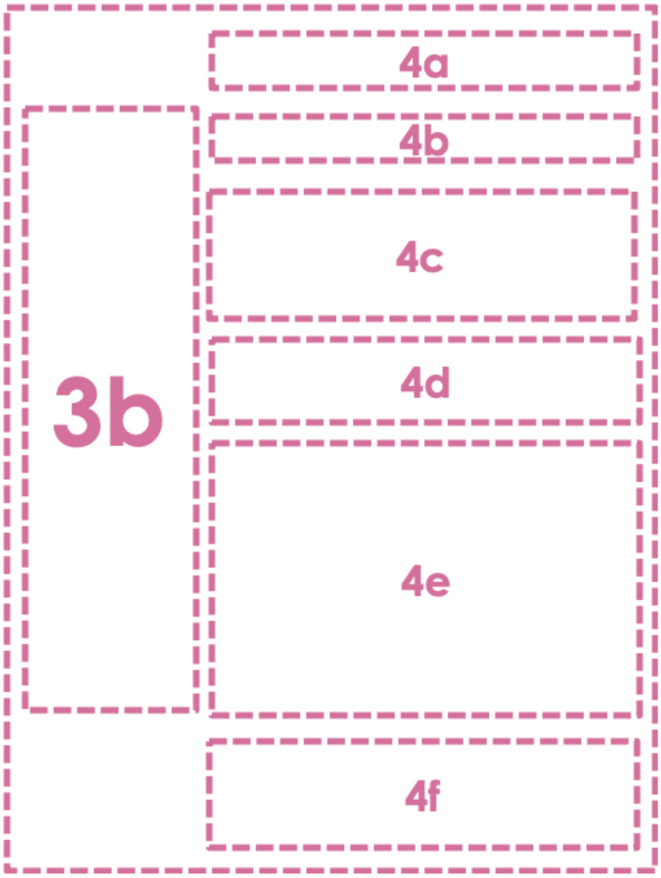


Figure 3: Example of an article page in a wireframe

Page name (4a)

The page name is used as the main heading for all page types. This is a key accessibility issue as it connects the page title to its name which is important, especially for users with assistive technology (a confirmation that the link I clicked on is leading me to the content I want). It is also important for search engines which are indexing the pages by the page name.

Block area (4b)

An accessible dynamic block area will be placed high on the page for the authors to use the page type with variation. The author can choose to use it for accessible videos, images or text, depending on the message.

Page image (4c)

Many article pages use a central image to reflect the content. This area can be used for accessible page images. The page image will have an ALT-tag connected to it.

Preamble (4d)

The accessible preamble will be prototyped as a separate text feature, connected to a specific CSS-class, to make it consistent when it comes to design as well as programmatically determined for assistive technology.

Free text content area (4e)

The WYSIWYG editor is made for free text content creation, including formatting of body text, headings, links and listings etc. The objects will be connected to individual CSS-classes to ensure the design pattern is consistent when it comes to type fonts, sizes etc. It is not possible to alter the editor in the template, but the prototype will be slightly modified, for example, by taking out italics as they may cause readability problems.

Block area (4f)

To provide the possibility for the author to add a function as the last part of the article page, the prototype will include an accessible block for listing or forms.

6.3.1. List of WCAG success criteria covered in the article page template

General WCAG criteria that can be affected by authors using the article page template
Success Criterion 1.1.1 Non-text Content (Level A)
Success Criterion 1.3.1 Info and Relationships (Level A)
Success Criterion 1.3.2 Meaningful Sequence (Level A)
Success Criterion 1.3.3 Sensory Characteristics (Level A)
Success Criterion 1.3.4 Orientation (Level AA)
Success Criterion 1.4.4 Resize text (Level AA)
Success Criterion 1.4.5 Images of Text (Level AA)
Success Criterion 1.4.10 Reflow (Level AA)
Success Criterion 1.4.12 Text Spacing (Level AA)
Success Criterion 2.4.2 Page Titled (Level A)

Success Criterion 2.4.4 Link Purpose (In Context) (Level A)
Success Criterion 2.4.6 Headings and Labels (Level AA)
Success Criterion 3.1.2 Language of Parts (Level AA)
Success Criterion 3.2.4 Consistent Identification (Level AA)
Success Criterion 4.1.3 Status Messages (Level AA)

6.4. Calendar event page

The calendar event page is used to create and communicate events on the website. This template prototype will consist of features to create individual and accessible start and end times, as well as the location of for the event.

The accessible features of the calendar event template selected for prototyping and include the following functions:

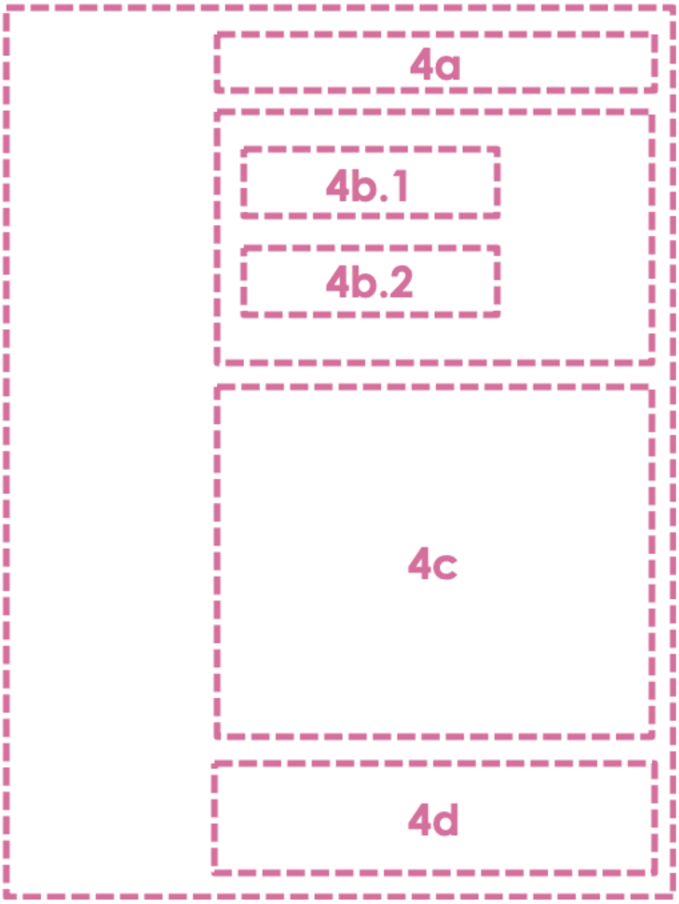


Figure 4: Example of calender

Page name (4a)
See description for article page.

Metadata of event (4b)

The prototype will use metadata for the event, to make sure the information is indexed in search engines and can be used when filtering and structuring of content. Start, end and location (4b.1-2) of the event will be made using a design pattern so that the content is easily recognisable to the end user.

Content area (4c)

Potential areas and blocks will be similar, or indeed the same, as for the article page.

Block area (4d)

The prototype will contain a block area to be used for an accessible forms block for registration to the event.

6.4.1. List of WCAG success criteria covered in the calendar event page template

General WCAG criteria that can be affected by authors using the calendar event template
Success Criterion 1.1.1 Non-text Content (Level A)
Success Criterion 1.3.1 Info and Relationships (Level A)
Success Criterion 1.3.2 Meaningful Sequence (Level A)
Success Criterion 1.3.3 Sensory Characteristics (Level A)
Success Criterion 1.3.4 Orientation (Level AA)
Success Criterion 1.4.4 Resize text (Level AA)
Success Criterion 1.4.5 Images of Text (Level AA)
Success Criterion 1.4.10 Reflow (Level AA)
Success Criterion 1.4.12 Text Spacing (Level AA)
Success Criterion 2.4.2 Page Titled (Level A)
Success Criterion 2.4.4 Link Purpose (In Context) (Level A)
Success Criterion 2.4.6 Headings and Labels (Level AA)
Success Criterion 3.1.2 Language of Parts (Level AA)
Success Criterion 3.2.4 Consistent Identification (Level AA)
Success Criterion 4.1.3 Status Messages (Level AA)

6.5. Page listing block

Accessible listings are very useful and common on all kinds of websites. The prototype page listing block will provide the author with possibilities to choose accessible design patterns for the heading, images, preamble, format, size and date for publishing.

The accessible features of the page listing block template selected for prototyping and testing include the following functions:

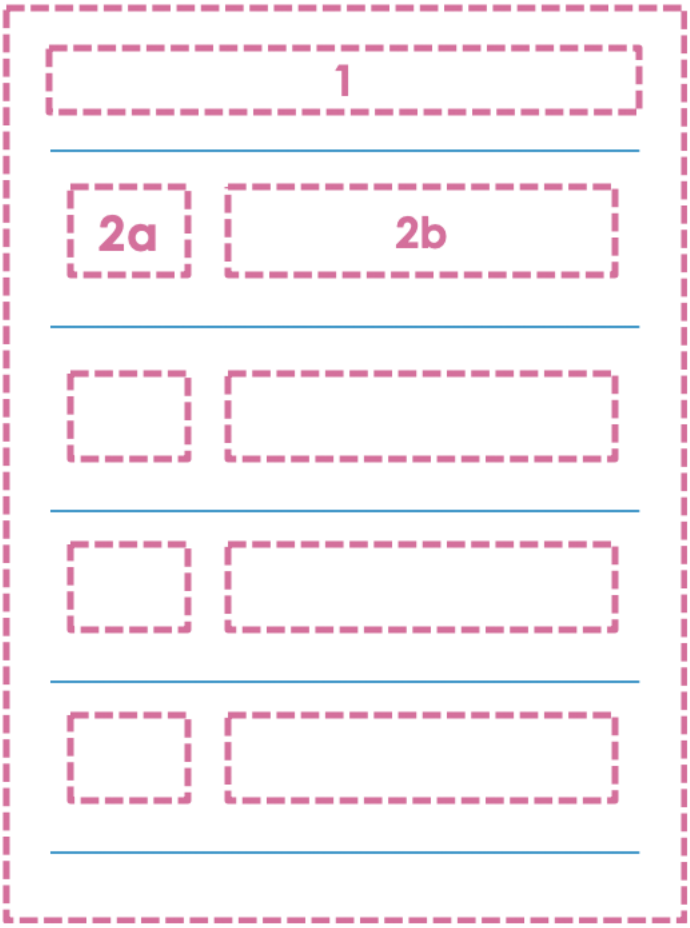


Figure 5: Example of listing blocks in a wireframe

Heading (1)
The listing will have a heading.

Listing (2)
Each page is represented by a separate row in the listing, the layout of which is configured by the preferences in the block. In the figure above, (2a) represents the area where the image of the listed page is placed and (2b) the area where the heading and preamble is shown. Each row in the list is linked to the corresponding page.

Preferences
In the prototype, manual preferences for the number of rows the list should contain and how it should be sorted will be provided.

6.5.1. List of WCAG success criteria covered in the page listing block

Specific WCAG criteria for the page listing block

Success Criterion 1.3.1 Info and Relationships (Level A)
Success Criterion 1.3.2 Meaningful Sequence (Level A)
Success Criterion 1.3.3 Sensory Characteristics (Level A)
Success Criterion 1.3.4 Orientation (Level AA)
Success Criterion 1.4.10 Reflow (Level AA)
Success Criterion 1.4.12 Text Spacing (Level AA)
Success Criterion 1.4.13 Content on Hover or Focus (Level AA)
Success Criterion 2.1.1 Keyboard (Level A)
Success Criterion 2.1.2 No Keyboard Trap (Level A)
Success Criterion 2.1.4 Character Key Shortcuts (Level A)
Success Criterion 2.4.3 Focus Order (Level A)
Success Criterion 2.4.4 Link Purpose (In Context) (Level A)
Success Criterion 2.4.6 Headings and Labels (Level AA)
Success Criterion 2.4.7 Focus Visible (Level AA)
Success Criterion 2.5.2 Pointer Cancellation (Level A)
Success Criterion 3.1.2 Language of Parts (Level AA)
Success Criterion 3.2.1 On Focus (Level A)
Success Criterion 3.2.2 On Input (Level A)
Success Criterion 3.2.3 Consistent Navigation (Level AA)
Success Criterion 3.2.4 Consistent Identification (Level AA)
Success Criterion 4.1.1 Parsing (Level A)
Success Criterion 4.1.2 Name, Role, Value (Level A)
Success Criterion 4.1.3 Status Messages (Level AA)

6.6. Calendar listing block

The calendar listing is necessary to provide an overview of the events of the calendar. The prototype calendar listing block will provide the option of usage on a standard page for single calendar events or a folder with several calendar events. The author is free to choose accessible design patterns for the heading, images, preamble, format and size of the event, as well as set the date for publishing.

The accessible features of the calendar listing block template selected for prototyping and testing include the following functions:

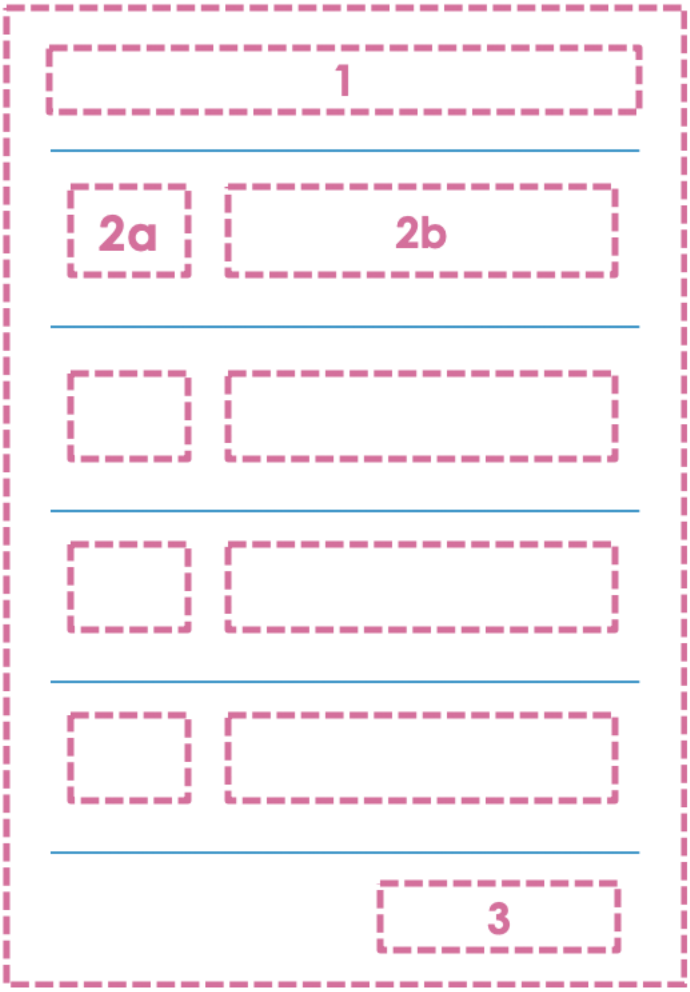


Figure 6: Example of calander listing block in a wireframe

Heading (1)
The calendar listing will have a heading.

Listing (2)
Each calendar event is represented by a separate row in the listing, the layout of which is configured by the preferences in the block. In the figure above, (2a) represents the area where the start-and end time of the calendar event is placed and (2b) the area where the heading, preamble and place is shown. Each row in the list is linked to the corresponding calendar event.

Link (3)
In the prototype, it will be possible for the author to add a link to a page where all calendar events are presented, in a compilation or an archive.

Preferences
In the prototype, the author will be able to choose the number of calendar events the list should contain, how it should be sorted and whether there should be a link to the compilation or archive. The default setting is to only show calendar events that have not yet occurred. This can be overridden by the author.

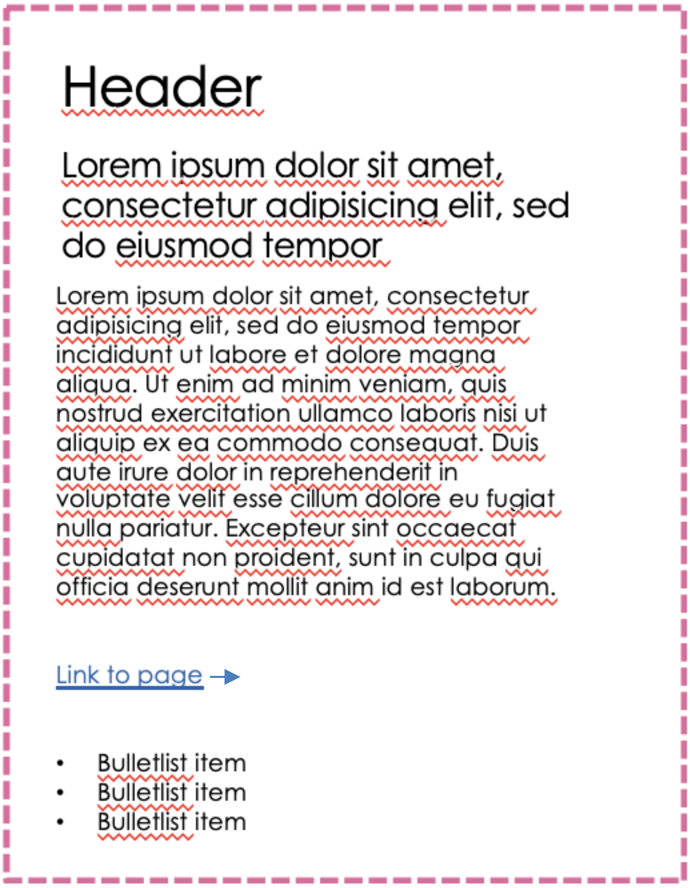
6.6.1. List of WCAG success criteria covered in the calendar list block template

Specific WCAG criteria for the calendar list block template
Success Criterion 1.3.1 Info and Relationships (Level A)
Success Criterion 1.3.2 Meaningful Sequence (Level A)
Success Criterion 1.3.3 Sensory Characteristics (Level A)
Success Criterion 1.3.4 Orientation (Level AA)
Success Criterion 1.4.10 Reflow (Level AA)
Success Criterion 1.4.12 Text Spacing (Level AA)
Success Criterion 1.4.13 Content on Hover or Focus (Level AA)
Success Criterion 2.1.1 Keyboard (Level A)
Success Criterion 2.1.2 No Keyboard Trap (Level A)
Success Criterion 2.1.4 Character Key Shortcuts (Level A)
Success Criterion 2.4.3 Focus Order (Level A)
Success Criterion 2.4.4 Link Purpose (In Context) (Level A)
Success Criterion 2.4.6 Headings and Labels (Level AA)
Success Criterion 2.4.7 Focus Visible (Level AA)
Success Criterion 2.5.2 Pointer Cancellation (Level A)
Success Criterion 3.1.2 Language of Parts (Level AA)
Success Criterion 3.2.1 On Focus (Level A)
Success Criterion 3.2.2 On Input (Level A)
Success Criterion 3.2.3 Consistent Navigation (Level AA)
Success Criterion 3.2.4 Consistent Identification (Level AA)
Success Criterion 4.1.1 Parsing (Level A)
Success Criterion 4.1.2 Name, Role, Value (Level A)
Success Criterion 4.1.3 Status Messages (Level AA)

6.7. Text block template

The accessible text block is meant to provide the author with the opportunity to create text content freely in any block area on a page. In the text block template prototype, it will be possible for the author to format the body text, headings, links and listings. The objects will be connected to individual CSS-classes to ensure the design pattern is consistent when it comes to type fonts, sizes etc.

The accessible features of the text block template selected for prototyping and testing include the following functions:



Header

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Link to page →

- Bulletlist item
- Bulletlist item
- Bulletlist item

Figure 7: Example of text block in a wireframe

6.7.1. List of WCAG success criteria covered in the text block template

General WCAG criteria that can be affected by authors using the text block template
Success Criterion 1.1.1 Non-text Content (Level A)
Success Criterion 1.3.1 Info and Relationships (Level A)
Success Criterion 1.3.2 Meaningful Sequence (Level A)
Success Criterion 1.3.3 Sensory Characteristics (Level A)
Success Criterion 1.3.4 Orientation (Level AA)
Success Criterion 1.4.4 Resize text (Level AA)
Success Criterion 1.4.5 Images of Text (Level AA)
Success Criterion 1.4.10 Reflow (Level AA)
Success Criterion 1.4.12 Text Spacing (Level AA)
Success Criterion 2.4.2 Page Titled (Level A)
Success Criterion 2.4.4 Link Purpose (In Context) (Level A)

Success Criterion 2.4.6 Headings and Labels (Level AA)
Success Criterion 3.1.2 Language of Parts (Level AA)
Success Criterion 3.2.4 Consistent Identification (Level AA)
Success Criterion 4.1.3 Status Messages (Level AA)

6.8. Image block template

The accessible image block prototype makes it possible to publish an image with an ALT-text and a description. The prototype will provide testing opportunities for making the ALT-text field mandatory or supportive.

The accessible features of the image block template selected for prototyping and testing include the following functions:

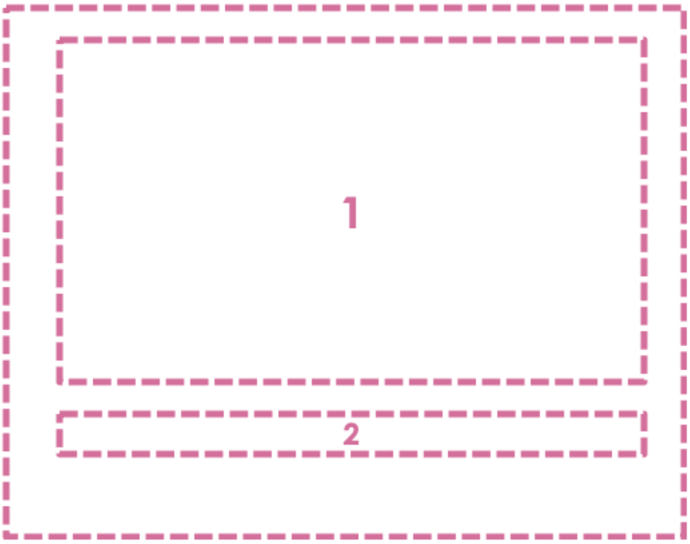


Figure 8: Example of image block template in a wireframe

Image (1)
This area is for placing the image in the block.

ALT-text (1)
The ALT-text is connected to the image to provide a text alternative to visually impaired users with screen readers.

Description (2)

Images usually need a written description also for sighted users. This is why the prototype will contain a field for describing the image in text that is shown to the user.

6.8.1. List of WCAG success criteria covered in the image block template

General WCAG criteria that can be affected by authors using the image block template
Success Criterion 1.1.1 Non-text Content (Level A)
Success Criterion 3.1.2 Language of Parts (Level AA)

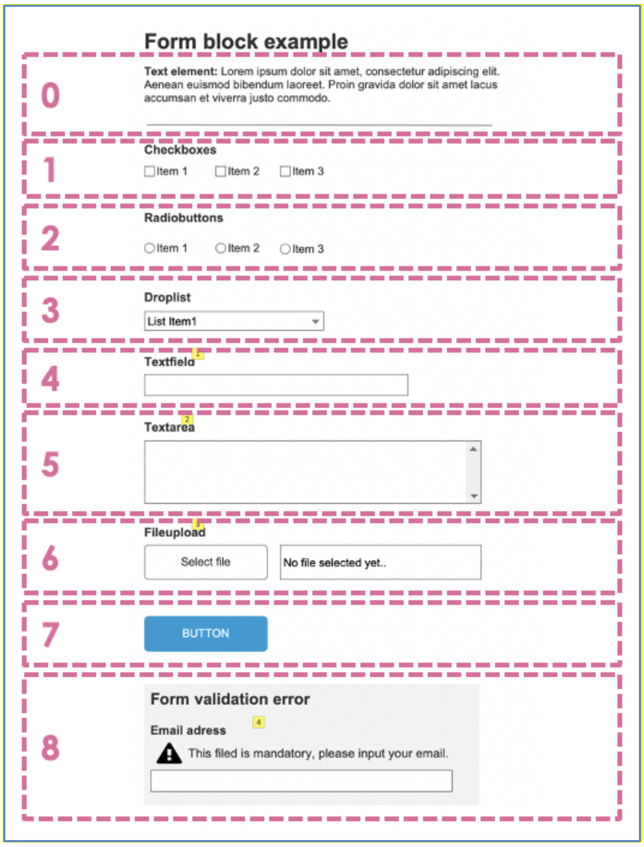
6.9. Forms block

Forms are generally one of the types of objects on websites where most accessibility problems arise. Forms can be anything from simple to complex, but the list of potential accessibility fails may be long even for a small form.

In the prototype, each form field will be connected to the corresponding heading, which makes it possible for users with assistive technology to understand where content is to be filled in.

The forms block will also have the possibility to make fields mandatory, and validating input data.

The accessible features of the forms block template selected for prototyping and testing include the following functions:



Form block example

Text element: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo.

Checkboxes

□Item 1 □Item 2 □Item 3

Radiobuttons

○Item 1 ○Item 2 ○Item 3

Droplist

List Item1

Textfield

Textarea

Fileupload

Select file | No file selected yet..

BUTTON

Form validation error

Email adress

This filed is mandatory, please input your email.

Figure 9: Example of forms block in a wireframe

Check boxes for multiple-choice response (1)

Radio buttons for A/B choices (2)

Drop list forlist items (3)

Text field for single row content (4)

Text editor for longer free text content, including the possibility to break rows, divide text into sections etc (5)

Document upload possibilities for attachments, including a button opening a dialogue view where the user can choose a document to upload. (6)

Submit button (7)

Validation to prevent the form to be sent in with errors. (8)

6.9.1. List of WCAG success criteria covered in the forms block template

Specific WCAG criteria for the forms block template
Success Criterion 1.4.1 Use of Color (Level A)
Success Criterion 1.4.3 Contrast (Minimum) (Level AA)
Success Criterion 1.4.4 Resize text (Level AA)
Success Criterion 1.4.10 Reflow (Level AA)
Success Criterion 1.4.11 Non-text Contrast (Level AA)

Success Criterion 1.4.12 Text Spacing (Level AA)
Success Criterion 1.4.13 Content on Hover or Focus (Level AA)
Success Criterion 2.1.1 Keyboard (Level A)
Success Criterion 2.1.2 No Keyboard Trap (Level A)
Success Criterion 2.1.4 Character Key Shortcuts (Level A)
Success Criterion 2.4.1 Bypass Blocks (Level A)
Success Criterion 2.4.3 Focus Order (Level A) (ID29)
Success Criterion 2.4.4 Link Purpose (In Context) (Level A) (ID30)
Success Criterion 2.4.6 Headings and Labels (Level AA) (ID32)
Success Criterion 2.4.7 Focus Visible (Level AA) (ID33)
Success Criterion 2.5.1 Pointer Gestures (Level A) (ID34)
Success Criterion 2.5.2 Pointer Cancellation (Level A) (ID35)
Success Criterion 3.2.1 On Focus (Level A)
Success Criterion 3.2.2 On Input (Level A)
Success Criterion 3.2.3 Consistent Navigation (Level AA)
Success Criterion 3.3.1 Error Identification (Level A)
Success Criterion 3.3.4 Error Prevention (Legal, Financial, Data) (Level AA)
Success Criterion 4.1.1 Parsing (Level A)
Success Criterion 4.1.2 Name, Role, Value (Level A)
Success Criterion 4.1.3 Status Messages (Level AA)

General WCAG criteria that can be affected by authors using the forms template
Success Criterion 1.3.5 Identify Input Purpose (Level AA)
Success Criterion 2.5.3 Label in Name (Level A)
Success Criterion 3.2.4 Consistent Identification (Level AA)
Success Criterion 3.3.2 Labels or Instructions (Level A)
Success Criterion 3.3.3 Error Suggestion (Level AA)

6.10. Video block

The accessible video block prototype will contain the option to publish a video in a block area with ALT-text and a description.

The accessible features of the video block template selected for prototyping and testing include the following functions:

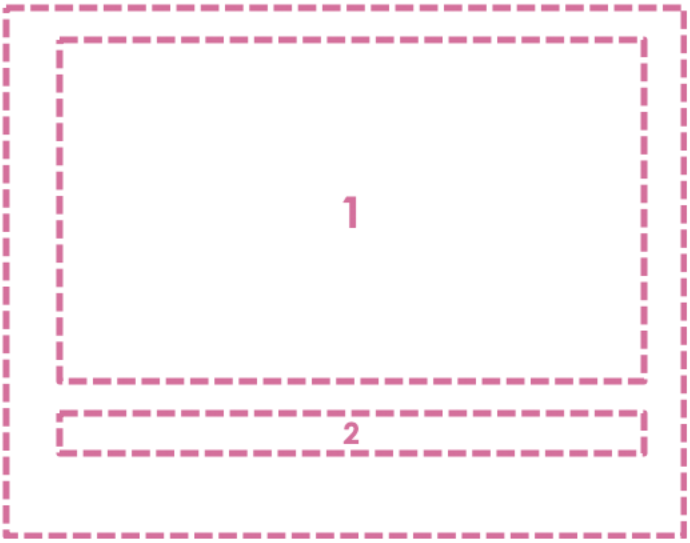


Figure 10: Example of video block in a wireframe

Video (1)
The prototype will provide an option to choose a link to a video from Vimeo or YouTube.

ALT-text for video (1)
An ALT-text will be connected to the video in the prototype, to provide information for users with screen readers.

Description (2)

The prototype will contain a possibility for the author to publish text in connection to the video to provide equivalent information to users with screen readers.

6.10.1. List of WCAG success criteria covered in the video block template

General WCAG criteria that can be affected by authors using the video block template
Success Criterion 1.1.1 Non-text Content (Level A)
Success Criterion 1.2.1 Audio-only and Video-only (Pre-recorded) (Level A)
Success Criterion 1.2.2 Captions (Pre-recorded) (Level A)
Success Criterion 1.2.3 Audio Description or Media Alternative (Pre-recorded) (Level A)
Success Criterion 1.2.4 Captions (Live) (Level AA)
Success Criterion 1.2.5 Audio Description (Pre-recorded) (Level AA)
Success Criterion 3.1.2 Language of Parts (Level AA)

7. Conclusion

The selected features to be prototyped and tested in WP2 cover a wide range of objects that web authors need in order to create a basic information website. They consist of items that are known to cause trouble for authors trying to create accessible content. They also tend to create barriers for end users with disabilities when the attempts to create accessible content fail. The combination of complete page templates and smaller block templates make the selection flexible and usable in a variety of situations and contexts. Built-in accessibility by default is abundant which should minimise the risk of technical accessibility issues arising. The features where manual handling is still needed will be thoroughly tested with web authors with varying previous knowledge of accessible content creation.