ROSKILDE UNIVERSITY

INFORMATICS

IT ARCHITECTURE AND USER-DRIVEN SOFTWARE DESIGN

Integration and automation of standard-compliant tagging

Authors

Daniel Șerbănescu
dase@ruc.dk
Student no.64513

Frederik Zeilberger Thulstrup
frzeth@ruc.dk
Student no.58223

Lukas Kucerik
kucerik@ruc.dk
Student no.66892

Anton Kjær Hansen
akjaerh@ruc.dk
Student no.64510

Tomas Nemecek
nemecek@ruc.dk
Student no.64509

Supervisor
Magnus R.P. Hansen

Teachers
Magnus R.P. Hansen
Keld Bødker
Niels Jørgensen
Troels Sune Mønsted

May 26, 2019

IN COOPERATION WITH

KRISTELIGT DAGBLAD

Contents

Part I

Reflection

Introduction

This reflection is a review of our project and explores some of the things that we learned throughout this process. This document comprises of four core areas covered by the course syllabus. Each area contains learning points, which describe our accomplishments, our failures and things that we could have done better.

Chapter 1

System Design Development

1.1 Challenges with scope definition

Nick Rozanski and Eoin Woods point out that a "Clear definition and agreement of scope are vital early milestones of any system development project." [RW12, p. 248] Ideally, according to Rozanski and Woods, the scope will be defined prior to project initiation [RW12, p. 248], and they further note, that if it is not defined, it may be done by oneself "...based on input from [the] stakeholders." [RW12, p. 248].

We struggled with not having a clearly defined scope from the beginning, due to the fact that Kristeligt Dagblad has multiple large processes each containing dozens of systems. Therefore, we were guided towards one of Kristeligt Dagblad's core business areas: the editorial process. Having a somewhat defined scope, which arguably was not very clear, provided us with a sort of dilemma: Should we ask the stakeholders to initially provide us with a clearer definition of scope? Or would a better approach be to start with an as-is analysis of the editorial process, and then agree on a clear scope definition, based on the analysis and stakeholder inputs, later on in the process?

We chose the latter, which to some extent caused a large part of the project to be very similar to a long iteration 0 of an inception phase within iterative and incremental software development processes [WIK19a].

This is partly why a large part of our project consists of analysis and system design.

The notion that we had to decide only between two options on how to approach a more clear scope definition, is a somewhat false dilemma since another option was present: finding another company with a clearly defined case.
It is captivating to imagine how completely different the project would have unfolded have we had a clearly defined scope and case. However, we believe that this option would have deprived us of the lesson of understanding why a precise and agreed upon scope, early on in the project, carries such significance.

1.2 What makes a good requirement?

A recurring discussion throughout the project revolved around the ambiguity in relation to what statutes a "well-defined" requirement. Due to the fact that this project's actual problem specification was crystallized rather late in the process, many of the stakeholder meetings were spent on as-is analysis. This made the stakeholders unable to be active in the iterative process of requirement specification, thus making it hard to know if the level of detail was accurate or not. Stakeholder engagement, when specifying (or extracting) requirements, can serve as a point of reference in i.e. figuring out whether the balance between a requirement description is either too

broad or too restrictive. But looking elsewhere for indicators of what makes a “well-defined” requirement could perhaps aid the stakeholders in their decision-making process.

Rozanski and Woods define a requirement as a stakeholder concern that is both specific, unambiguous and measurable [RW12, p. 106]. This definition is somewhat broad and vague, which the authors also note. This, in turn, caused our requirements to also be broad and vague, and due to the lack of proper stakeholder feedback, made us spend tedious time and effort to narrow the requirements down. However, in the process of defining the requirements too precisely, our architectural document grew into more of a software development document instead of a general architectural guideline.

In the end, we are not sure what the better approach to our situation would have been. We could have pushed stakeholders more in our meetings for more specific problematic situations, which could lead to more concrete requirements from early on. However, from our understanding, without our specification of all the problematic situations within the editorial process, stakeholders themselves did not fully realize them. Therefore, they would not have been able to provide us with clear requirements even if prompted to do so.

1.3 Characteristics of an effective AD

Rozanski and Woods clarify that "[a]n effective AD must balance seven desirable properties: correctness, sufficiency, timeliness, conciseness, clarity, currency, and precision." [RW12, p. 198] In this section we will discuss how we attempted to balance these properties in our AD.

Correctness and Sufficiency
The most efficient way to judge correctness is to reflect on two key questions: Did the AD "...correctly represent the needs and concerns of [the] stakeholders."? [RW12, p. 199]. And were we able to "...define an architecture that will meet those needs" [RW12, p. 199]? Unfortunately, we did not engage sufficiently with the stakeholders which makes this aspect challenging to accurately assess. We delve into this issue in the *The importance of stakeholder participation* section.

Sufficiency relates to the key question of whether the AD contained *enough* information. Although, there is ambiguity to what exactly defines "enough", the most effective way to judge Sufficiency is, like with the Correctness property, by engaging with the stakeholders. We believe, that we were able to find a somewhat appropriate balance, in regards to the descriptive text throughout the AD, as we managed to significantly cut down on superfluous text without losing any crucial information.

Precision and Conciseness
Much like the point above, Precision, relates to an AD that "...contain[s] sufficient information to allow the system’s architecture to be understood" [RW12, p. 205], and at the same time "...provide[s] a solid basis for analysis, design, and implementation." [RW12, p. 205]

We believe that we became more effective in finding this balance as the project went on, for example:

- transitioning from ordinary text to tables, when presenting detailed information

- a strategy for managing and referring to elements of the AD (i.e. requirements, viewpoints & perspectives and diagrams)
- the use of diagrams instead of long descriptive text
- balancing between abstraction and precision.

One of the biggest inefficiencies throughout the project is related to the Conciseness property. We simply spent too much time on details that either ended up being revised or completely removed. This misspending of time occurred when writing descriptive text and modelling detailed scenarios, which were ultimately outside of our scope. This issue could have been minimized if we had defined a clear strategy for modelling and writing at project initiation, which is detailed further.

Timeliness
As pointed out above, we could have been more productive, in e.g. modeling, if we have had a clear strategy from the beginning. A part of such a strategy could have mitigated our tendency to follow the "Big Modelling Up Front (BMUF)" anti-pattern [AMa] in contrast to doing Just-in-time (JIT) modelling, or "model storming" as it is also known [AMb].

Similarly to the other processes, throughout the span of this project, we became more competent at properly using JIT-modelling.

Clarity
The Clarity property relates to the fact that "...each stakeholder group should be able to understand those parts that are relevant to them." [RW12, p. 203]. As we did not engage with the stakeholders in a sufficient manner, it is challenging to evaluate our AD from the perspective of this property. However, as this project has certain academic constraints, we consider two types of collaborators, which cannot be classified as "project stakeholders", but whom we will classify as "academic stakeholders". These are our reviewers and our instructors.

During review sessions, we received valuable feedback from the academic stakeholders. This feedback focused primarily on the understandability of the written content as well as the models. We were able to adjust the AD according to this feedback and considered the review sessions along with the academic counseling extremely helpful.

In contrast to the other AD attributes, we believe that, in the end, we have followed the Clarity principle fairly well. Despite our limited stakeholder engagement, when we presented them with our artifacts they were able to easily understand them.

Currency
The Currency property relates to keeping the AD up-to-date once the system is deployed [RW12, p. 204]. Since our system did not have the chance to be in production, this property was not our focus at this stage of the project. However, we did keep a change log of relevant changes to the AD, which arguably relates to the Currency property.

Overall, we were more competent in keeping the change log up-to-date at the beginning of the project compared to later on. While this fact did not critically affect the overall product, we would have saved some time during the writing of this reflection report, had we been consistent in keeping the change log up-to-date.

1.4 The art of modelling

One of the major problems during the AD creation was the fact that we created models and views for the academic deadlines before actually defining the problem and scope of our project. This led us to mismanage our time, as we focused on producing artifacts for the as-is analysis to keep up with the deadlines instead of the problem definition. Creating these models helped us immensely with understanding Kristeligt Dagblad's architecture, however, once we defined our focus, many of the models were out of scope, such as the concurrency view, which we initially thought we might need as a self-selected view. Other models also had to be reworked entirely as they were focusing on irrelevant parts of the editorial process.

This caused us a lot of frustration with following the AD template in the first two-thirds of the project. However, once we narrowed down the boundaries of the project scope, following the AD template became much easier as we were able to relate various views and perspectives to the project and thus understand their value.

For a future project, we learned that it is more beneficial to spend as much time as possible with the stakeholders at beginning of the project to clearly figure out the direction, instead of trying to blindly follow the course deadlines. Thus the other half of the project could be focused more on the production of academic artifacts, as that enables a better understanding of theory and easier artifact creation.

1.5 System Design Development Conclusion

The biggest takeaway from this area of the project is that the scope is the most important for a worthwhile project. Without it, it is automatically more challenging to create meaningful requirements and produce a valuable product. Once a scope is defined, it is easier to deliver correct and concise information both for the academic and business standards.

Chapter 2

Implementation and Change Management

2.1 The importance of stakeholder participation

There are a number of reasons why our access to the stakeholders was limited and it is clear that our strategy for stakeholder engagement was insufficient. An example of such insufficiency can be gathered from the *activity log*, which shows that all of the formal stakeholder meetings happened in the early stages of the project, before the actual scope definition. This is problematic as adequate stakeholder input bears a crucial role in requirement specification (as mentioned in the *What makes a good requirement?* section), and a failure to engage them will lessen the probability of a successful system.

We also engaged informally with the stakeholders after the scope was defined in a limited manner. This happened through Daniel, acting as a *Communicator* when questions arose. Although we did anticipate limited access to the stakeholders, as seen in the *Risk Matrix*, we should have put a greater effort into engaging the stakeholders in formal meetings throughout the entire span of the project and not only during the project's earlier stages. This ties back to our flawed prioritization of academic deadlines over problem definition as mentioned in the previous *section*.

In hindsight, a clear strategy for stakeholder engagement should have been defined, with emphasis on developing a process that would have allowed the stakeholders to participate in project planning.

At the beginning of the project, we were also insecure in our ability to bring value to Kristeligt Dagblad and saw our project as more of a burden on them. We, therefore, tried to do as much work as possible on our own, even though it would have been an immense help to have more meetings with the stakeholders. This had significant ramifications for the entire project as mentioned in multiple places throughout this reflection.

Despite the insufficient stakeholder engagement, we believe that we were successful in some areas of stakeholder communication. One of the areas was good preparation for meetings with the stakeholders. This preparation included a precise agenda for the meeting, as well as specific questions for each stakeholder. This allowed us to use our limited time with them efficiently and get the most valuable information.

2.2 Minimizing resistance to change

We were not able to sufficiently explore the topic of change management in our AD, because the project did not come to that stage. However, in this section, we examine and reflect on how change management strategies can be used in order to reduce resistance to change.

One way would be to ask the stakeholders about their opinion on how to make the change as smooth as possible. They could have identified issues that could arise during change and helped

us mitigate them. Unfortunately, in this case, we did not engage the stakeholders to try to proactively solve this and thus we explore other possibilities below.

Technochange is a technology-driven organizational change [Mar04]. Our project fits this description because the tagging process involves a change in both IT and organizational processes. Our project arose from a top-down decision, using the Commanding Change Strategy [PB08, p. 740], meaning that the decision came from the management. This can create obstacles for the system adoption by the users, as they were not involved in the decision-making process. To improve adoption, there are various approaches that can be taken, which are elaborated below.

2.2.1 IT change

The proposed system, covered throughout the AD, requires the journalists to adjust their work process, which will most likely cause resistance to the adoption of the system.

One way to change user behaviour would be to enforce article tagging. This could be done by requiring the article to be tagged before it is published. However, this would most likely cause annoyance and dissatisfaction among the journalists. Therefore they might do the minimum amount of work to get around the aforementioned requirement. This could result in inaccurate tags that would negatively impact the quality of the outcome.

To counter possible resistance to change, we advise implementing some functionality that is valuable for journalists and possible only due to tagging. This could, for example, take the form of a "profile page" for journalists, that shows what tags they use and thus their areas of expertise. This would allow journalists to see the value of the new process. By including the people that are affected by the change, the "resistance to change" is expected to decrease [Mar04, p. 5].

2.2.2 Organizational Change

Multiple changes might also be required at the organizational level, for the tagging system to be successfully adopted. As outlined in the section above, this can be done by implementing functionality to show the value of the process. However, this is not necessary as other strategies are available.

Using innovation attributes, we can identify potential pitfalls with system adoption. This will, in turn, help us determine the required extent of organizational change. The innovation attributes are Relative Advantage, Compatibility, Complexity, Triability and Observability [Rog83, chapter 3].

Relative advantage
Relative advantage is "[t]he degree to which the innovation is better than what is in place already." [MT16, p. 403]

We believe that journalists currently do not see much value in tagging and consider it as "extra work". This change does not benefit them directly but will provide business value. Convincing them of this could very well be a major obstacle for adoption.

Since it is difficult for the journalists to see the personal benefit of article tagging, we recommend a training process to be put in place to make the journalists understand the value of tagging for the company as a whole.

Compatibility
An innovation that is compatible with the users' existing processes will be easier to accept. [OU]

Tagging is at the end of the article editing process and can be done relatively fast. The tagging solution is integrated with the ARI, which is a system that the journalists use daily and does not require them to use an entirely new system.

We, therefore, do not expect large resistance due to compatibility.

Complexity
The tagging solution is designed to be easy to use. It suggests tags and the tagging standard is concise, meaning there is a limited number of options. This relieves the journalist from having to make complex choices. Thus we do not expect resistance because of the complexity.

Triability
One way to reduce potential resistance-to-change is to let the users try it before it is fully implemented. In the case of our tagging solution, we could introduce parallel running [LBP12, p. 63] during implementation. This would also increase Observability as described below.

Observability
Users are more likely to adopt new processes if they see other people using them successfully. At this stage of the project, we cannot accurately assess Observability as this is a new system at Kristeligt Dagblad that is meant to introduce tagging.

Analysis Result
With the discoveries gathered from identifying innovation attributes we ascertain the degree of organizational development.

Organizational development covers change methodologies that target managers' attitudes and behaviors, human resource development and training, organization culture, reward systems, job redesign, organizational structure, etc. [MW11, p. 378]

Based on our analysis, other than the relative advantage, we do not expect serious resistance to our system. Therefore, we see no need for large organizational developments. We suggest using human resource development and training to acclimate users to the new processes and establish why they are valuable.

2.3 Implementation and Change Management Conclusion

Similar to the other sections, the main learning point is that proper stakeholder participation is of the utmost importance. Putting more effort to get stakeholder feedback on the change management strategies, would have been the best use of our resources as it has a direct impact on the adoption rate.

Chapter 3

Evaluation of IT

The biggest issue with Evaluation of IT within this project was the insufficient interaction with the stakeholders in regards to proof-of-concept & prototypes, UI designs and summative evaluations. In the following section, we outline what we managed to accomplish and what we could have done better in accordance with the course theory.

3.1 The value of a proof-of-concept and a prototype

After the first few discussions about the architectural changes that would need to be performed in order to accommodate our tagging solution, we became aware of the reality that the performance of this solution would play a key role in such a system. Generally, every process that has to deal with text analysis or extraction of meaning from a context is a time- and resource-consuming matter. This could introduce a serious issue for user-acceptance of our solution or in case of performance improvement would place a heavy burden on Kristeligt Dagblad's ecosystem.

We developed a proof-of-concept [RW12, p. 224] for a keyword extractor using one of the candidate solutions. We created a simple console application in Python which uses the multi rake algorithm [Vit19]. We used this application for a simple baseline performance estimates elaborated upon further in the *Performance Evaluation* section.

In the future we would develop other simple proof-of-concept applications using other solutions such as using the tf-idf [WIK19b] ranking, TextRank algorithm [MT04], topic analysis using machine learning [ML] or external service providers. Further, we would develop a proof-of-concept for the Tag Mapper. The Keyword Extractor and the Tag Mapper are separate systems and rely on the same interface. Thus connecting all the keyword extractor solutions to the mapper would be a relatively simple process. This would allow us to easily test and compare the candidate solutions for the keyword extractor.

Having developed these proof-of-concepts and performed benchmark tests, we would approach Kristeligt Dagblad for formal reviews and structured walk thorough as according to Rozanski and Woods [RW12, p. 219]. Having access to the company's system architect would allow the company to consider all the technical pros and cons of the different solutions and make an informed decision based on concrete data and proven evaluation techniques.

3.1.1 Performance Evaluation

The aforementioned proof-of-concept of the Keyword Extractor was used to estimate performance, to test its capabilities, to find out its limitations and to evaluate a risky element of the to-be architecture. The results of the evaluation point to a possible solution, however, we identified two aspects which will affect the performance of the consumable solution [AL12, p. 29]:

the process of mapping the extracted keywords to tags and the reality that the keyword extraction is required to support Danish. This could prove to be an obstacle since a majority of text analysis tools are tailored to English.

Even though we were able to make the initial performance estimations, we did not communicate these results to our stakeholders. We focused too much on finding different candidate solutions to this problem. It might have been more useful to present the prototype to Kristeligt Dagblad immediately after measuring the results. This could have helped align our vision of the system with theirs and speed up the decision-making process on how to proceed further.

3.1.2 Design and User Experience Evaluation

In the Usability perspective of our AD, we used heuristics to evaluate the proposed UI. The heuristics are inspired by generally agreed-upon characteristics of a usable product [Kom]. Although UIs should be developed using Participatory Design [SR13, resume], we used heuristic evaluation because we were unable to engage the stakeholders in time.

If we were to use Participatory Design, the benefit of mock-ups would increase as the stakeholders would provide us with valuable feedback. In this case, we would focus especially on journalists to make sure that the UI does not disturb their existing workflow.

3.1.3 Summative Evaluation

Eventually, we would have liked to perform a summative evaluation [Jyt16, p. 46] of the chosen solution, to see whether the system achieved it is objectives. Such evaluation would be performed after a piloting phase [Wel13], meaning the system would be a part of the journalists' daily processes while it is still being improved. This evaluation would include interviews with:

- Kristoffer, the software architect - to compare the estimated impact of the tagging solution with its actual impact on the company's architecture
- The journalists and editors - to see the level of satisfaction with the system and whether it has not made the process of article editing inconvenient
- Mads Emil, the Head of Digital Department and Hans Christian, Chief Revenue Officer - to find out whether the tagging solution helped to unify and categorize the articles as it was initially intended
- The marketing team - to see whether the unified tagging eased the advertisement process or increased the website traffic to related articles and potentially helped increase revenues.

3.2 Evaluation of IT Conclusion

In conclusion, proof-of-concept applications, prototypes, and mock-ups provide significant value to the project, so that all the stakeholders and our team are in agreement in regards to the solution. However, in our case, the value was diminished by the lack of communication with the

stakeholders. Nevertheless, this issue would not have been as significant, if the project was given more time which would allow for more stakeholder engagement.

Chapter 4

IT security

The security risks within the editorial system are not concerned about personal data but ensuring that the company business data is safe. Personal data is handled in another department of Kristeligt Dagblad. This allowed us not to concern ourselves with personal data regulations, such as the GDPR [EU].

4.1 Addressing ARI security weaknesses

Although the *Security Perspective*, found in the Appendix, was developed before the project scope was clarified. There are parts of it that influence the to-be system, because they are related to the ARI. This concerns the access to the ARI in relation to confidentiality and integrity [SB18, p. 25]. To summarize the *main points*, we believe that the ARI could benefit from additional and improved layers of security. The Tag Interface is a sub-module of the ARI, therefore the tagging solution is directly affected by the upper layer of security. To mitigate the weaknesses of the inherited security, the tagging solution would benefit from its own layer of security.

Additionally, one of the requirements was portability and the ability of the tagging solution to integrate with various Drupal versions, it was a false assumption that we could rely primarily on inherited security.

With the previous points in mind, we realized that our to-be views should have been revised to reflect a stronger focus on the security perspective. For instance, we should have considered adding an authentication module to the component diagram in the functional view and added the notion of how to protect data at rest and in transit in the information view.

4.2 Insufficient prompt for security

One of the reasons for us not having any security requirements is because we believe that we did not properly prompt the stakeholders on topics related to security. Therefore, we may have missed out on identifying more problematic situations in this area. This fact, along with the stakeholders' increased focus on functionality, caused us to not properly take security into account.

In an ideal scenario, we should have asked for stakeholder security concerns in regards to the tagging solution i.e. after the scope has been defined. This could possibly have led the project in another direction. With their concerns about security, this would allow us to implement the security into the to-be architecture.

4.3 IT security conclusion

Despite our knowledge in the area of software development and that security should be embedded into the design of any product by default, we still managed to overlook security as one of the main aspects of the project.

In the future, we would like to focus on security even if the project presents itself as not having strong security requirements. This could be done by collaborating with the stakeholders to derive their security needs.

Part II

Architectural Description

Change Log

Version	Date	Changes
0.01	21 Feb.	Project Intiation. Gantt Chart.
0.02	28 Feb.	Project Charter. Stakeholders Analysis v.1
0.03	03 Mar.	Revision of Project Charter and Stakeholder Analysis.
0.04	05 Mar.	Context view draft. Revision of Project Charter.
0.05	12 Mar.	Editorial process was updated with more detailed information.
0.06	14 Mar.	Updated Context View with the Feeds Entity. Initial work on Functional View. Added Risk Matrix to project charter.
0.07	17 Mar.	Updated functional view and the context view. Initial work on information view.
0.08	21 Mar.	Updated both functional diagrams and information diagram.
0.09	31 Mar.	Updated section about the identified problems and solutions. Updated stakeholder analysis. Initial concurrency diagram.
0.10	02 Apr.	Updated project charter. Added text and model on concurrency view. Mention concurrency issues in the concurrency view introduction.
0.11	09 Apr.	Restructured the document.
0.12	11 Apr.	Implemented AD Template
0.14	18 Apr.	Initialized to-be architecture
0.15	20 Apr.	Added to-be context and to-be information views
0.16	23 Apr.	Formatted for review
0.17	25 Apr.	Removed concurrency view Created scenarios and Requirements
0.18	27 Apr.	Reworked diagrams to proper notation Created introduction out of project charter
0.19	04-06 May	Restructured document to decrease page number
1.0	07 May	Restructured and front-loaded the report
1.1	09 May	Cutting down the project
1.2	14 May	Merged stakeholder tables together Fixed up Functional and information view Added to-be section
1.3	20 May	Finished front-loading the report
1.4	25-26 May	Finished the AD

Table 4.1: Table of changes made in the project report.

Chapter 5

Introduction

Kristeligt Dagblad is a well-known player in the Danish newspaper market which has gone through a long process of changes and evolution since its establishment in October 1896. The company is focusing on topics such as faith, ethics and existence and strives to put current issues into a larger context through objective journalism.

Kristeligt Dagblad's biggest issue is the lack of a tagging system. Kristeligt Dagblad previously had a manual tagging system, however the inconsistency in article tags made the system impractical and therefore it was dropped in 2017. In this report we propose a tag suggestion system to remove subjectivity from the tagging process.

Kristeligt Dagblad has print media as one of the biggest sources of revenue. However, over 600 000 active readers access the digital platform monthly. Kristeligt Dagblad therefore wants to improve the user experience of the website as much as possible, and they believe that a better topic tagging system would satisfy this.

5.1 Scope

Our focus is on the editorial process, which will serve as basis for the as-is architecture. The following tasks are performed in the editorial process:

- Manual input of articles by journalists.
- Automatic import of articles from internal and external systems.
- Update of existing articles.
- Publishing and un-publishing of articles.
- Exporting articles to newsletters and other feeds.
- Converting articles to audio format.
- Presenting articles to readers.

The to-be architecture will revolve around consistent article tagging based on keyword extraction from the content of the article.

5.2 Project Status

The current state of this project is that we detailed the architectural proposal for the tagging system. The next steps for this project would be to present this document to Kristeligt Dagblad and create prototypes for possible solutions, test and compare them, and see how scalable they are. The company would then decide on which prototype they would like to develop further.

Afterwards, in collaboration with the company, we would create a strategy on how to introduce this system into production and their daily processes.

Glossary

Term	Definition
AD	AD refers to an Architectural Description document.
ARI	ARI is an acronym for the danish Artikelredigeringsinterface which means Article Editing Interface.
Editorial process	A process comprising of a chain of activities that lead from article creation to article publication and content sharing on various media.
Editorial system	Editorial system is a part of editorial process, comprising of ARI and Drupal.
IPTC	International Press Telecommunications Council developing and publishing Industry Standards for the exchange of news data of all common media types.
Keyword Extractor	The part of the tagging solution that analyses the article content and extracts relevant keywords from it.
Software architecture	Refers to the fundamental structures of a software system and the discipline of creating such structures and systems.
Tag Interface	The part of the tagging solution that the journalists directly interact with.
Tag Mapper	The part of the tagging solution that is responsible for mapping the extracted keywords from the article's context to the final tags.
Tagging Solution	The system for suggesting and assigning tags to the news articles that our group has worked on during this project.

Table 5.1: Dictionary of terms and their explanation

Chapter 6

System Stakeholders and Requirements

In this section, we identify various stakeholder groups relevant to the project, their main concerns and derived requirements.

The proposed to-be architecture will only contain solution to one of the identified problematic situations, consult section 11.6 in the Appendix for the full list.

6.1 Stakeholders

Stakeholder group	Stakeholders roles & names	Concerns
Acquirers pay for the system and allocate resources to each department.	**The board of directors**, using Hans Christian Kock and Kristoffer Arrild as proxies.	Effective use of resources.
Assessors oversee that the system follows the regulations	**Head of Digital Development**, Mads Emil Lidegaard	The systems and tools used must abide to legislative regulation and follow the standards of the market.
Communicators handle documentation and training for the systems.	**IT Support**, led by Jørgen Schmidt provides assistance for journalists in regards to the systems.	ARI can be improved to minimize the amount of training required.
Developers create and update the systems.	**Web developers** and development consultants such as Carsten Dan.	Saxo-Databridge is central to the editorial process. New additions should not hinder data processing.
Maintainers develop and fix the system.	This task is performed by the web developers and the server administrator.	The systems require a fair amount of documentation in order to be easy maintainable.
Production Engineers are responsible for the deployment.	The person responsible for this is the server administrator.	The system's components must be tested and ready for production before the deployment starts.
Suppliers provide parts of the infrastructure for the editorial process.	They provide support for systems like Saxo, VisioLink, Nota and Sentia.	Kristeligt Dagblad's requirements are important to understand to be able to provide the best service possible within the boundaries of the contract agreements.
Support Staff help employees to use the systems.	This role is fulfilled by **IT support** and ad-hoc by the other users of the systems.	Support staff must be announced when any major change is applied to the systems.
System Administrators keep the system running.	The **IT department** with Jørgen Schmidt as the lead is in charge of this.	The systems should be easy upgradeable and backwards compatible for upgrades to be as seamless as possible.
Users are the people who use the systems.	They are the **journalists**, **editors**, **page layout designers** and **the proof readers**. Represented by Julie Greve Bentsen and Hannibal Erngaard.	Editing of articles should be as painless as possible. Reliability and availability of the systems is very important. User experience should be one of the main focus points.
Software Architect orchestrates the systems' relations.	Kristoffer Arrild and our group are the architects for this project.	Data is a central to the company business and architecture should accommodate different platforms.

Stakeholders' roles and concerns

6.2 Requirements

Based on the stakeholder concerns we derived the need for an objective tagging system. Currently, there is no tagging system in place. Previously it was up to the journalists' judgment and was thus inconsistent. This was not optimal for marketing purposes and findability of related articles. Our tag suggestion solution would create consistent tagging within a limited tag scope. Below in table 10.7 a list of requirements categorized by system qualities.

ReqID[1]	Req. Name	Description
Usability Perspective		
FR-1	Tag suggestion and selection	The journalist sees a list of suggested tags and selects from them
FR-2	Tag relevance	The suggested are contextually relevant to the article and ordered by their relevance
FR-3	Tag correction	The journalist can select a tag from a list of standardized tags, when none of the suggestions match the article.
NR-1	Limited suggestions	Depending on the article, the number of suggested tags is not too low nor too high.
NR-2	Visibility	The suggestions are clearly visible on the screen, near the area of main article content. No scrolling is involved. The tagging progress is reported back to the client.
Regulation Perspective		
NR-3	Tagging standard	The suggested tags are based on global tagging standards such as IPTC[2]instead of being maintained by the company
Evolution Perspective		
AR-1	Portability	The tagging system is technology agnostic and can be used in different software environments
AR-2	Input & Output	Tagging system receives the article text and outputs suggested tags
AR-3	Low-coupling and Integration	Tagging system integrates with ARI and is decoupled from the rest of the editorial system ensuring high cohesion
Performance and Scalability		
NR-4	Performance	The tag suggestion process should not take any longer than a few seconds to process for a Kristeligt Dagblad article sized text
NR-5	Concurrency	The tagging system should be able to handle multiple concurrent requests without slowing down
AR-4	Scalability	The tagging system should be able to scale easily
Availability and Resilience Perspective		
AR-5	Error handling	If an error occurs, the system does not crash but reports the error
AR-6	Non-blocking behavior	The tag suggestion process does not block other activities
Localization Perspective		
NR-6	Tag language	The tags must be in Danish. The tagging standard used must be translated if needed.

[1] FR = "functional requirement", NR = "non-functional requirement, AR = "architectual requirement"
[2] IPTC - International Press Telecommunications Council defines global standards of the news media.

Table 6.1: Overview of requirements

6.3 System Scenarios

After we made a clear overview of the requirements that our to-be architecture should fulfill, we can identify how the system should meet these requirements by writing system scenarios.

The sections below describe the functional scenarios (concerned with the functional features of the system) and system quality scenarios (describing how the system reacts to certain changes in its environment).

6.3.1 Functional Scenarios

Scenario id	FS-1
Overview	The system performs text analysis on the written article content and produces a set of terms that are relevant in the article context. The terms are then mapped to the closed set of tags defined by the tagging standard. The system then presents the matching tags to the journalist writing the article.
External Stimulus	Journalist has pressed a button to suggest the tags for the written article.
System Response	The system suggests a set of tags relevant to the article.

Table 6.2: The "suggest tags" scenario

Scenario id	FS-2
Overview	The system should present the tags to the user in an order based on relevance.
External Stimulus	The user have submitted the content of an article for tag suggestion.
System Response	The system returns a list of suggested tags based on relevance. While the exact definition of relevance is yet to be defined, one suggestion could be to base relevance on analysis of existing articles that shares the submitted articles context.

Table 6.3: The "tag relevance order" scenario

Scenario id	FS-3
Overview	The system recognizes the tags selected by the journalist and saves them together with the article. If none of the suggested tags fit properly the journalist can select a tag from the full standardized tag set.
External Stimulus	The journalist has selected the tags which they found most relevant to the article from the suggested tag set and has saved or published the article.
System Response	The system responds in the same way as before when saving or publishing the article.

Table 6.4: The "save tags" scenario

Scenario id	FS-4
Overview	The system disables the functionality for suggesting the tags in a current article after it has already started the process. The functionality becomes available again after the system is finished with suggesting the tags.
External Stimulus	The journalist chooses to suggest tags for an article.
System Response	Once the journalist chooses to suggest tags for an article, the button becomes disabled and the system is not accepting more requests for suggesting tags in the current article until the tag suggestions appear on the screen.

Table 6.5: The "too many requests" scenario

6.3.2 System Quality Scenarios

Scenario id	NS-1
Overview	The tag suggestion process should not last longer than a few seconds and should not block any other journalist's actions.
Environment Changes	The journalist chooses to suggest tags for an article.
System Behaviour	It will not last longer than a few 8 seconds between the journalist's action of performing the tag suggestion and the actual suggestions being visible on the screen. During this time, the journalist can perform other tasks in the UI except for saving or publishing the article.

Table 6.6: The "performance" scenario

Scenario id	NS-2
Overview	The system displays an error message when the tags could not be suggested or the suggestion process is taking too long.
Environment Changes	The journalist chooses to suggest tags for an article.
System Behaviour	If an error happens in the tag-suggestion process, the system will clearly inform the journalist about this state. Similarly, if the suggestion process takes more than a few seconds, the system notifies the journalist in a similar manner.

Table 6.7: The "error occurred" scenario

Scenario id	NS-3
Overview	The system displays progress to the user while the tag suggestion process is ongoing.
Environment Changes	Changes to the ARI are necessary to support implementation of progress display.
System Behaviour	When an article is submitted for generating tag suggestions, progress of the process should be displayed to the user. One approach is to use a generic loading spinner, but detailed progress could also be displayed if the tag suggestion system reports when certain conditions have been met.

Table 6.8: The "visible progress" scenario

Scenario id	AS-1
Overview	The system has little or no knowledge of the environment in which it exists causing low interdependence between the components of the existing architecture.
Environment Changes	Components of the existing architecture are changed, removed or added.
System Behaviour	Since the tagging system are not bound to components of the existing architecture, modification or removal of said components would not have an effect of the tagging system. Modification to the tagging system would, in the same way, not have an effect of the existing architecture.

Table 6.9: The "Low-coupling" scenario

Scenario id	AS-2
Overview	The tag suggestion system is able to fetch the article heading, sub-heading and body of an article.
Environment Changes	There might not be changes to the environment.
System Behaviour	The tag suggestion system could fetch an article title, lead, kicker and body directly from the database. An unique ID for the article would need to be submitted as input data to the tag suggestion system. Another approach is to send the article title, lead, kicker and body as part of the request to the tag suggestion system. The ARI would in this case need to include a function to grab, format and send the data to the tag suggestion system.

Table 6.10: The "Input-data" scenario

Chapter 7

Architectural forces

7.1 Goals

The goal of the project is to provide three things. They concern the tagging solution described previously:

- Create a tagging system that has a limited set of tags and is effective at the same time
- Make tagging as automated as possible while providing a possibility of manual control
- Be able to find related articles and track tags

7.2 Constraints

One of main constraints is that the tagging system solution should be resource effective and straightforward to implement.

The solution should not require large-scale architectural changes to the existing systems. Only the ARI and the database is expected to change.

7.3 Evolving Architecture

Kristeligt Dagblad is a company that has had its IT solutions worked on for many years. Their architecture is mature and complex. However, as we learned in the beginning of our collaboration, the company is currently in the process of changing its software architecture from Drupal 7 to Drupal 8. Our proposed solution is decoupled from the other systems, which means their architectural change does not affect the outcome of this project.

7.4 Architectural principles

> "*An architectural principle is a fundamental statement of belief, approach, or intent that guides the definition of an architecture. It may refer to current circumstances or to a desired future state.*" [RW12, p. 134]

Principle Reference	AP-01
Principle Statement	The implemented tags should follow an existing standard.
Rationale	Kristeligt Dagblad does not have the ability and manpower to maintain a tagging standard.
Implications	The taxonomy structure must be thought of and maintained by an external entity.

Table 7.1: Principle AP-01

Principle Reference	AP-02
Principle Statement	The new architecture should be portable.
Rationale	Kristeligt Dagblad is in the process of moving architecture and needs to easily port architecture components from a software environment to another.
Implications	The proposed architecture needs to have as few dependencies as possible to other systems.

Table 7.2: Principle AP-02

Principle Reference	AP-03
Principle Statement	The new tagging solution needs to work on danish articles.
Rationale	The company has danish-only articles and its user base comprises of danish-speaking readers.
Implications	Keyword extraction solutions that do not work on danish text would be discarded.

Table 7.3: Principle AP-03

Chapter 8

Architectural Views As-Is

The following models and architectural descriptions are going to describe the as-is architecture of Kristeligt Dagblad's system

8.1 Context View

"*The Context view of a system defines the relationships, dependencies, and interactions between the system and its environment — the people, systems, and external entities with which it interacts*."[RW12, p. 248]

8.1.1 Model

Figure 8.1 presents the context model at abstraction level 1 in the as-is state of the architecture of the editorial system.

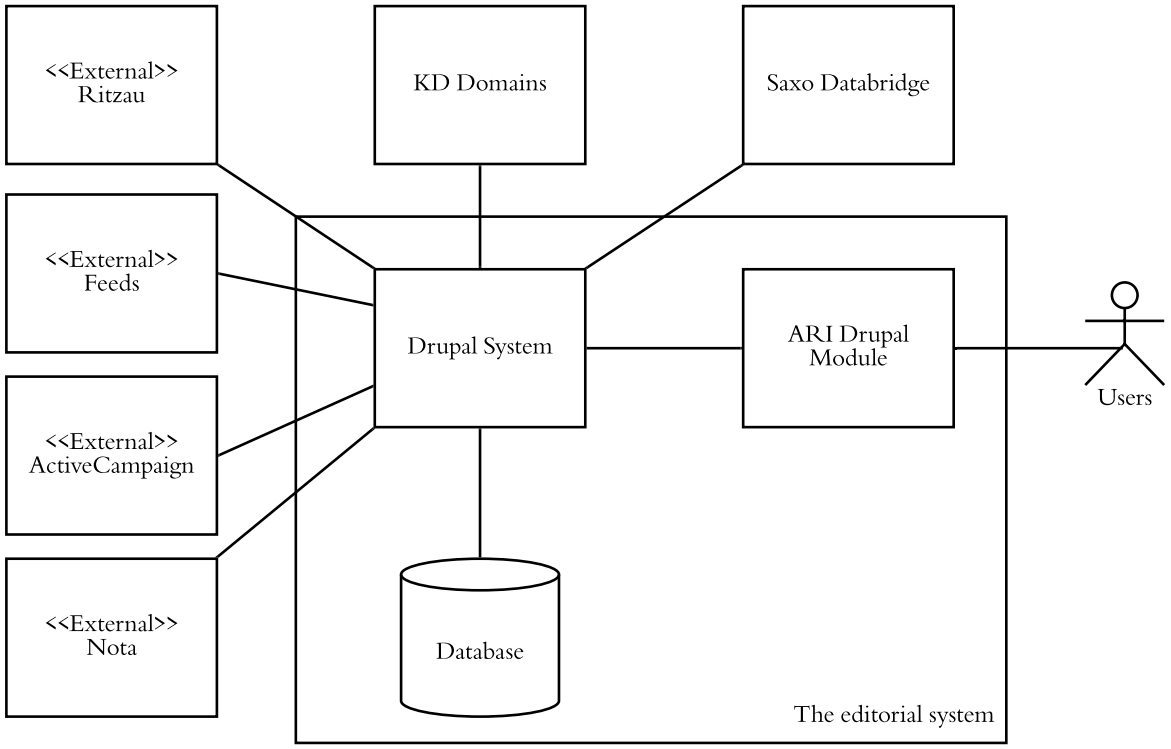

Figure 8.1: Level 1 Context model of the Editorial System (As-Is).

8.1.2 The Editorial System

The Editorial system's main responsibilities are handling article input, outputting the articles to relevant systems and storing articles. The system is comprised of:

- The ARI Drupal Module - an article editing interface, which is one of the ways for journalists to input their articles
- Drupal system - the back-end backbone of this system, connecting other systems together
- Database - internal article storage

For more information about systems outside the editorial system refer to section 11.9 in the Appendix.

8.1.3 Users

Users that interact with the editorial system are content creators, editors, developers and service maintainers.

8.2 Functional View

"*[The functional view] documents the system's functional structure—including the key functional elements, their responsibilities, the interfaces they expose, and the interactions between them.*."[RW12, p. 267]

8.2.1 System Activities

Articles can be created in three ways. It is either created in an external Saxo system and edited in ARI, created directly in the ARI, or ARI retrieves an article from Ritzau, another external system.

To see the external Saxo-related part of the editorial process refer to section 11.9 in the Appendix.

Below in figure 8.2 an UML activity diagram shows the process from article creation in ARI to publication.

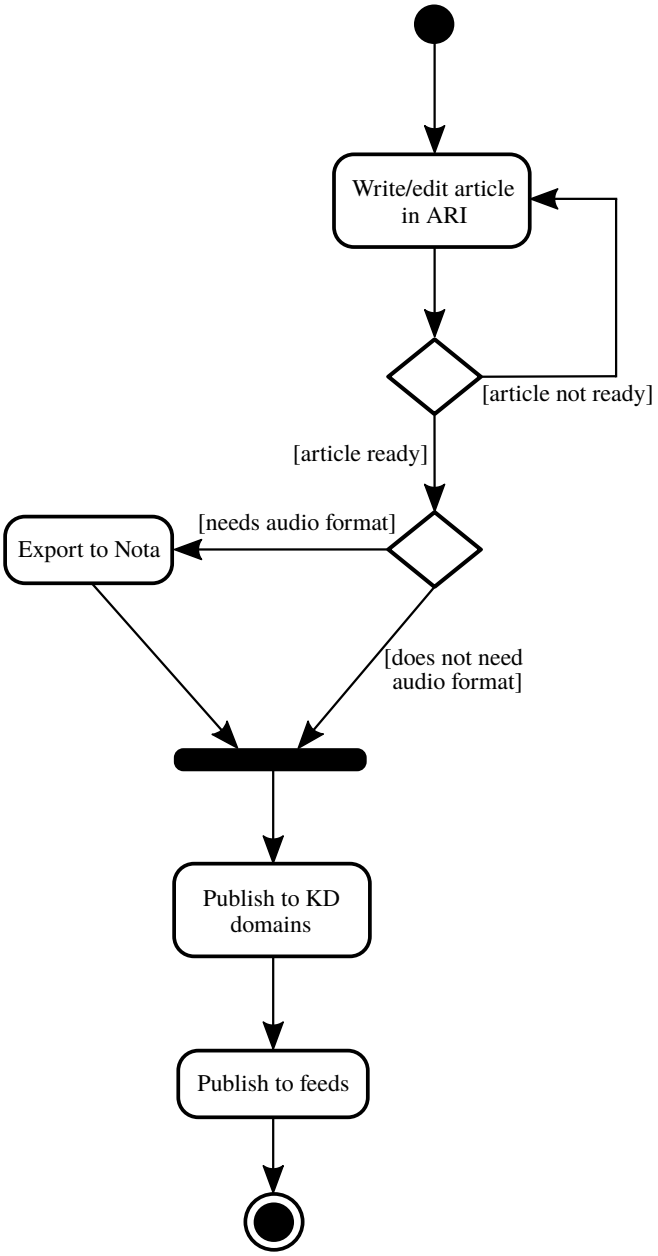

Figure 8.2: Activity diagram of the ARI-related section of the editorial process

The ARI system is built on top of Drupal, but for our analysis purposes ARI and Drupal are considered separate. Articles written in the ARI can only target the website. If an article should be turned into an audio format, it is exported to Nota, an external service that returns an audio file.

Afterwards it is then published to one of the Kristeligt Dagblad domain websites, which then inform external feeds.

Figure 8.3, presented below, shows the article publishing process when the article from Ritzau is fetched by ARI. The process is the same as described in the activity above, however Ritzau articles will never be turned into an audio format.

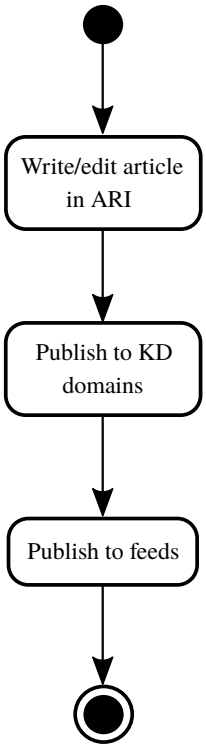

Figure 8.3: Activity diagram of Ritzau article retrieved by ARI

8.3 Information view

"*You use the Information view to answer, at an architectural level, questions about how your system will store, manipulate, manage, and distributes information.*"[RW12, p. 294]

8.3.1 Information Structure and Content

Below in figure 8.4 a level 0 ER diagram is presented showing the different data entities and their relationships.

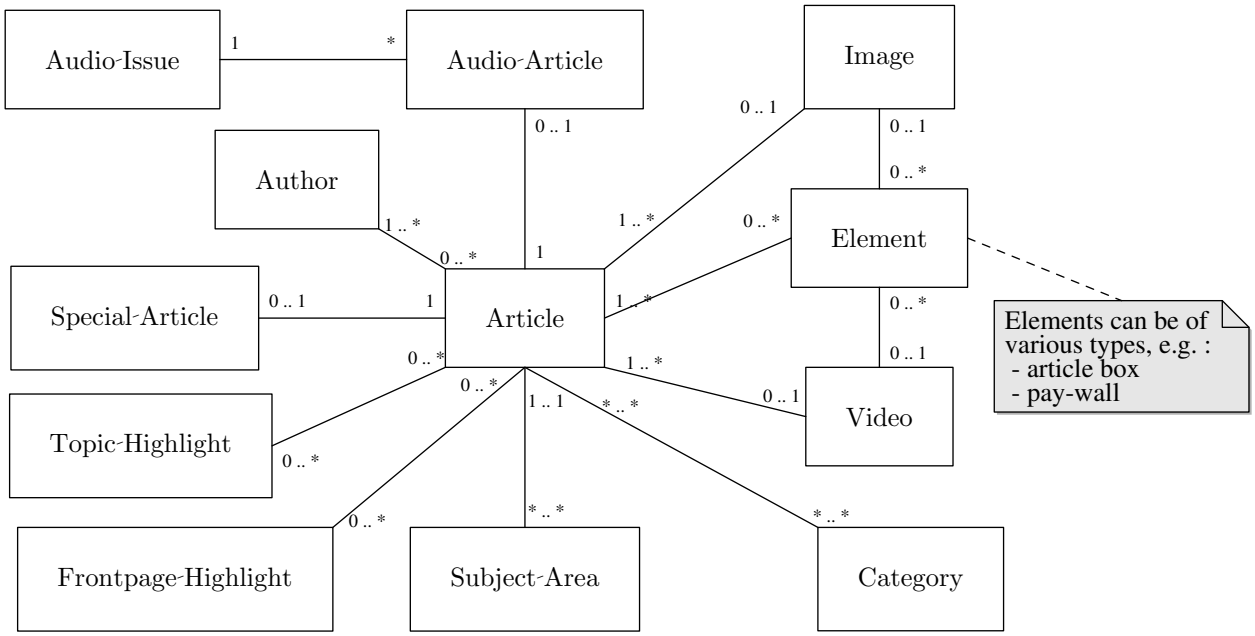

Figure 8.4: ER diagram level 0

Article and its relationships are the central entities of the editorial process.

However since our solution will only add new entities, we only focus on the article. For an overview of the remaining entities, refer to 11.7 in the Appendix.

Below in figure 8.5 a level 1 ER diagram is shown only focusing on the Article entity with its attributes.

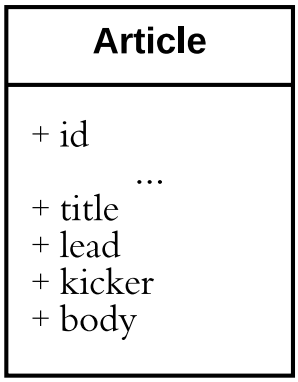

Figure 8.5: ER diagram level 1 showing the Article entity with attributes

An article is defined by its title, lead, kicker and body. Lead is the first paragraph of an article introducing the topic and also contains short conclusion. The kicker is a sentence long teaser for the article when it is featured on the front page.

8.3.2 Information Ownership

Below in table 8.1 we provide a information ownership [RW12, p. 319] grid for the relevant data items in the existing architecture. We identified only three different classes of information handlers:

- Owner - holds the definitive value for the data item
- Reader - can read but not change the data item
- Updater - can modify existing data items

	Article	Audio Article	Author	Special Article	Topic Highlight	Frontpage Highlight
ARI	Updater	Reader	Reader	N/A	N/A	N/A
Drupal	Owner	Owner	Owner	Owner	Owner	Owner

Table 8.1: Information Ownership Grid

Chapter 9

Architectural Views To-Be

The following models and architectural descriptions are going to describe the to-be architecture of Kristeligt Dagblad's system

9.1 Context View

9.1.1 Scope

The scope of the to-be context view is limited to the topic tagging entities within the editorial process. This happens within the writing and publishing of digital articles.

9.1.2 Context Model

As opposed to the as-is context model, the to-be context model shows only the relevant entities that will play a role in article tagging. Below in figure 9.1 is a level 1 model focusing only on the editorial system.

The two new systems additions are the Tag Mapper and the Keyword Extractor (marked with gray). The Keyword Extractor can be both custom made or be a third party solution.

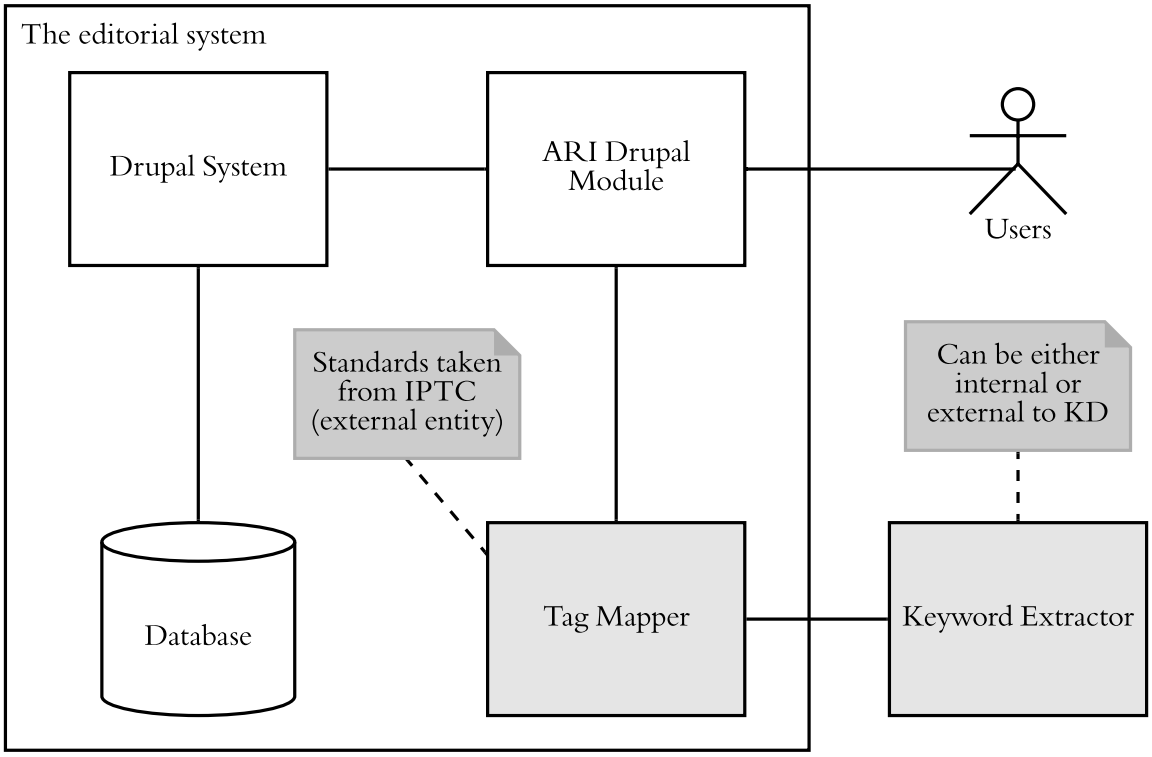

Figure 9.1: Level 1 Context view of the Editorial To-Be System

The Keywords Extractor's responsibility is to extract relevant keywords from the article's text. The keywords are sent to the Tag Mapper which maps keywords to tags. The tags are a subset of the IPTC NewsCodes standard, to read more about it and our implementation turn to Table 10.7

9.1.3 Meeting Requirements

Below in table 9.1 is a description of how the requirements are met in relation to the context view.

Req. ID	Req. Name	Fulfillment of Requirements
NR-3	Tagging Standard	The tags are taken from the IPTC standard and stored in the Tag Mapper
AR-1	Portability	The additional systems are independent of the editorial system and therefore portable
AR-3	Low-coupling and Integration	The tag mapper exposes only one interface for the ARI and therefore integrates into the editorial system
AR-4	Scalability	Since the tag mapper and keyword extractor are external entities to the editorial system, they can scale easily

Table 9.1: How the requirements are met in relation to the context view

9.2 Functional View

The Functional view to-be focuses on how well our proposed tagging system interacts with existing systems. Below in figure 9.2 a component diagram can be seen which describes the different components along with super- and sub components.

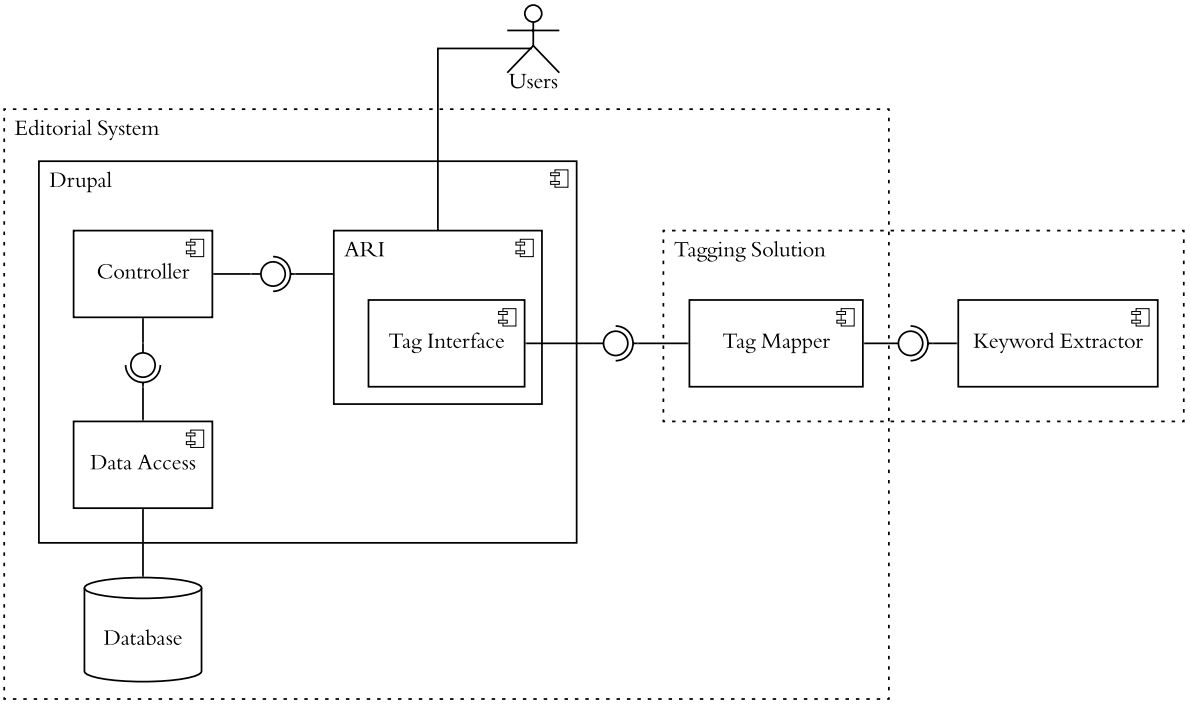

Figure 9.2: Component Diagram of the To-Be Architecture

The users interact with the Tag Interface through the ARI.

The controller and data access components, in the model above, are abstractions for modules that are not a relevant part of the editorial process. They are used to point out that ARI does not directly interact with the database.

9.2.1 Functional elements

The tables below describe the different components from figure 9.2. Each component has their own responsibility along with specification of their inbound and outbound data.

Component	Drupal
Responsibilities	The backend of the entire system and data access
Interface – Inbound	Article raw data
Interface – Outbound	Article presentation data

Component	ARI
Responsibilities	Interface for article editing
Interface – Inbound	Article Raw data
Interface – Outbound	Article processed data

Component	Tag Interface
Responsibilities	Presents tag suggestions
Interface – Inbound	Article data
Interface – Outbound	Suggested Tags

Component	Tag Mapper
Responsibilities	Maps article keywords to IPTC NewsCode root topics
Interface – Inbound	Article data: Heading, Lead, Kicker, Body
Interface – Outbound	Tags

Component	Keyword Extractor
Responsibilities	Extracts keywords from text
Interface – Inbound	Text
Interface – Outbound	Keywords

9.2.2 Meeting Requirements

Below in table 9.2 is a description of how the requirements are met in relation to the functional view.

Req. ID	Req. Name	Fulfillment of Requirements
FR-1	Tag Suggestion and Selection	The ARI is extended by the Tag Interface which is connected with the Tag Mapper and is responsible for presentation and selection of tags
FR-2	Tag Relevance	The Keyword Extractor performs a text analysis on the article content and generates relevance score for keywords which are further mapped to tags. The tags with a score are then sent to the Tag Interface
FR-3	Tag Correction	The Tag Interface presents the allowed tags and provides the option to add or remove the tags from the article
NR-1	Limited Suggestions	The Tag Mapper orders the tags by their relevance but only suggests the ones with the highest score
NR-2	Visibility	The Tag Interface makes sure that the Tag Mapper's status is always visible to the user
NR-3	Tagging Standard	The Tag Mapper maps the keywords to the IPTC subjects (sub-tags) and further to the IPTC root subjects (root-tags). The full set of IPTC subjects is used to make it easier to map keywords to a sub-tag and then map the sub-tag to a root-tag as it is defined in the IPTC standard.
AR-5	Error Handling	If an error occurs in the Tag Mapper or Keyword Extractor, the message is propagated to the Tag Interface to notify the user
AR-6	Non-blocking Behaviour	The Tag Interface does not block ARI activities

Table 9.2: How the requirements are met in relation to the functional view

9.3 Information view

9.3.1 ER diagram

Below in figure 9.3 a to-be level 1 ER diagram is presented which adds the new two entities with their attributes.

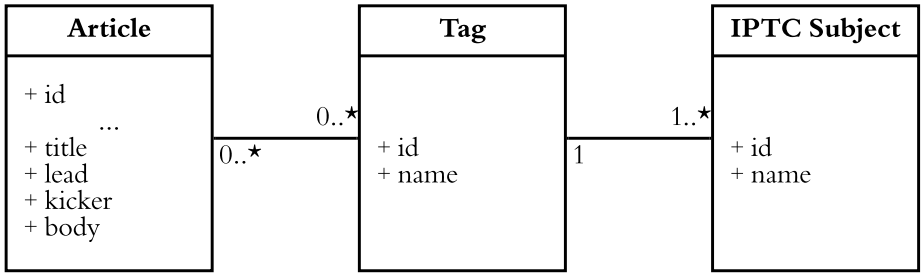

Figure 9.3: To-Be ER Diagram Level 1

Article
In the proposed solution, the article entity remains unchanged.

Tag
Tag is defined by its id and name, and is based on the IPTC Newscode standard. IPTC NewsCodes defines subjects hierarchically. As Kristeligt Dagblad is interested in tagging articles mainly for marketing purposes, only a subset of IPTC subjects is used as tags, namely the 17 root subjects. The subject code of the tag is used as id for the tag entity, however the name has been translated for Kristeligt Dagblad's purposes.

IPTC Subject
IPTC subject is an entity based on the IPTC NewsCode standard. It allows for an effective mapping of keywords, as all subjects have a clear relation to their root subjects and thus tags as well. The subject codes are used as ids and the names have been translated to Danish.

To see the set of tags please refer to 11.10.

9.3.2 Data flow

The figure 9.4 below shows the data flow of the tagging process.

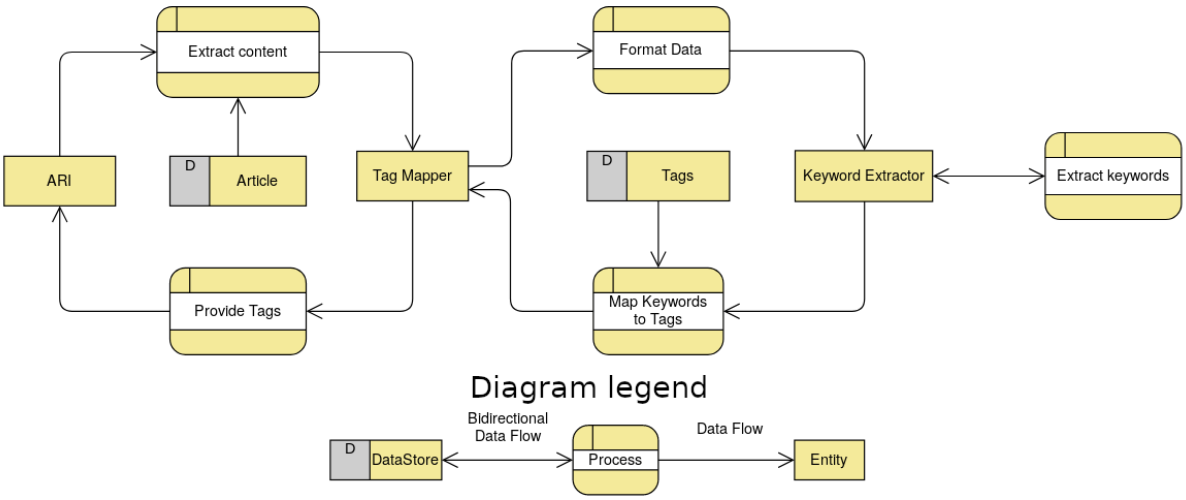

Figure 9.4: To-Be Data Flow Diagram Level 1

9.3.3 Timeliness and latency

Tags can be assigned at any point in time during the article creation process. Therefore the users should be informed about the timeliness of the tag data. The tags could become out of date by the point when the article is finished. So re-tagging of the article is needed for on-going stories. However the tags provided by the Tagging Mapper will never be stale as they are based on current input.

9.3.4 Meeting Requirements

Below in table 9.3 is a description of how the requirements are met in relation to the information view.

Req. ID	Req. Name	Fulfillment of Requirements
FR-1	Tag Suggestion and Selection	The tags are sent along with their scores from the Tag Mapper to the Tag Interface
FR-2	Tag Relevance	The tags are sent along with their scores to present relevance
NR-3	Tagging Standard	The Tag Mapper contains the information about the whole IPTC standard and the Tag Interface is only aware of the root tags of the IPTC standard
NR-6	Tag Language	The tag entity is stored together with its Danish equivalent in the Tag Mapper so that the Danish tag can be sent to the Tag Interface
AR-2	Input & Output	The Tag Mapper receives input in the form of article title, lead, kicker and body and outputs the tag suggestions with scores

Table 9.3: How the requirements are met in relation to the information view

Chapter 10

System Qualities

10.1 The Usability Perspective

"*The Usability perspective ensures that the people who interact with the system can work effectively.*"[RW12, p. 568]

10.1.1 User interface design

The solution has to cover all requirements and it is up to the developer ensure this. Below we present some mock-ups to help visualize a possible implementation within the ARI.

Figure 10.1 below shows the ARI with the Tag Interface marked with the blue square.

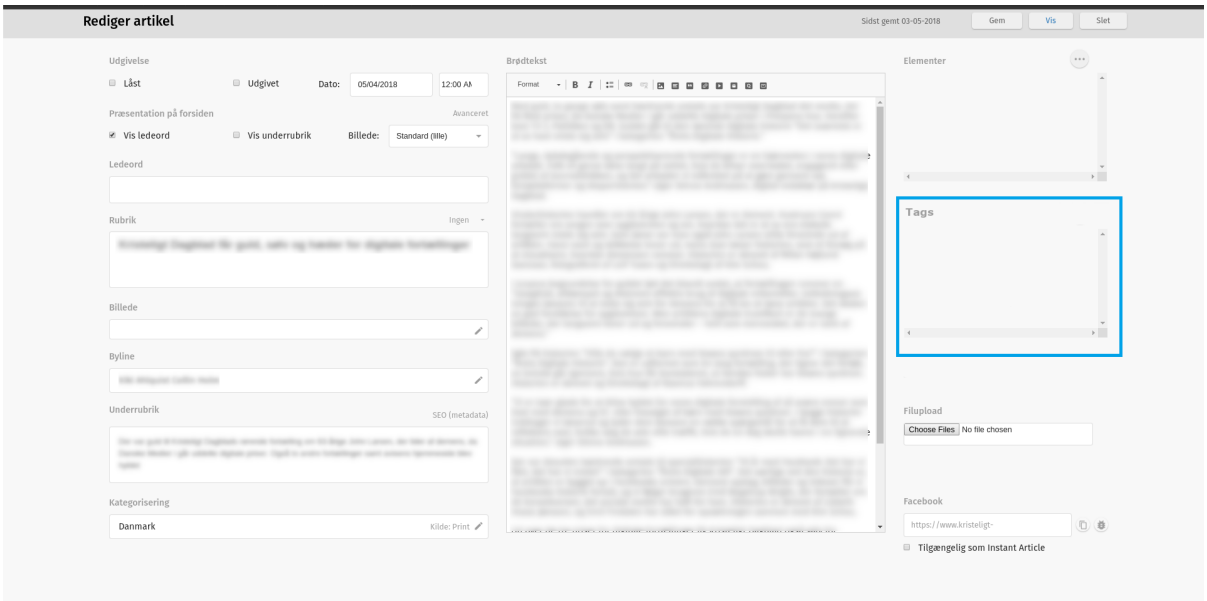

Figure 10.1: Overview of the ARI with the Tag interface

Below in figure 10.2 is a mock-up of the Tag Suggestion interface.

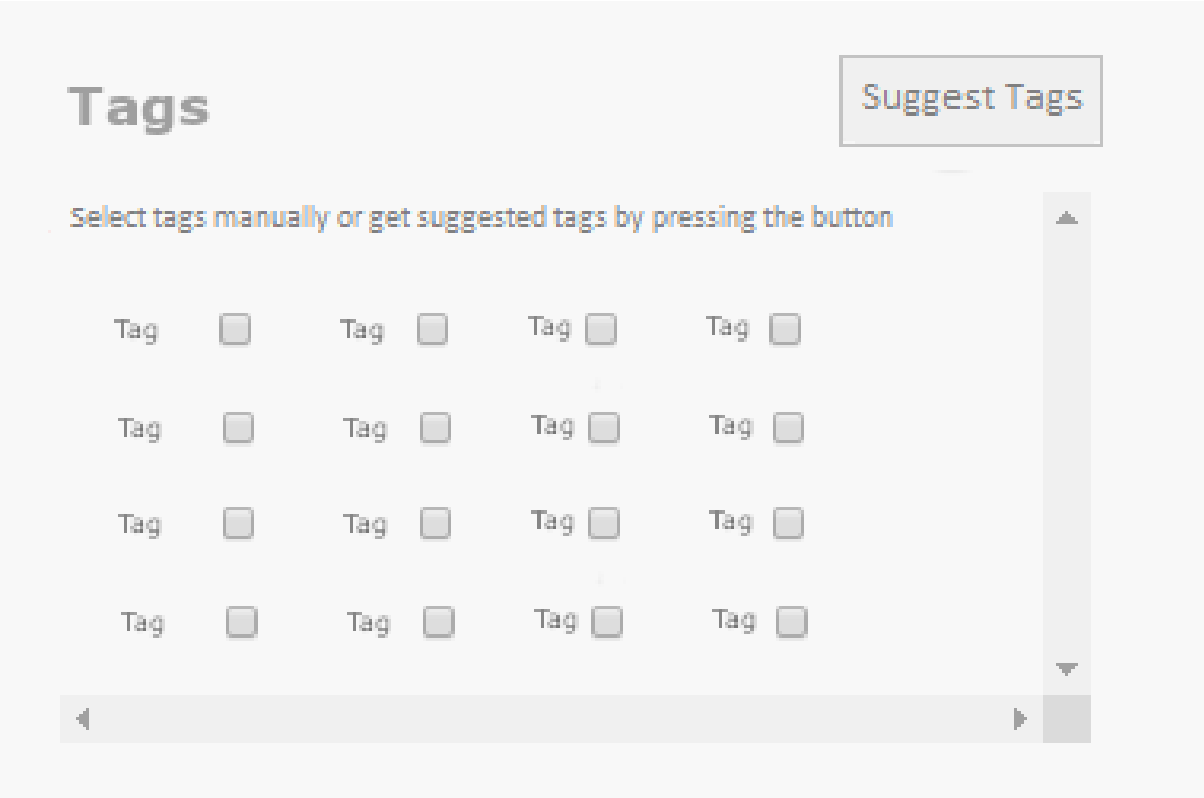

Figure 10.2: ARI tagging component before getting suggested tags

The journalist has the option to either get suggested tags or manually select tags if they want to. Once the button is pressed the suggest button cannot be pressed before the process is done. A stop button is present along with a loading indicator, as can be seen in figure 10.3.

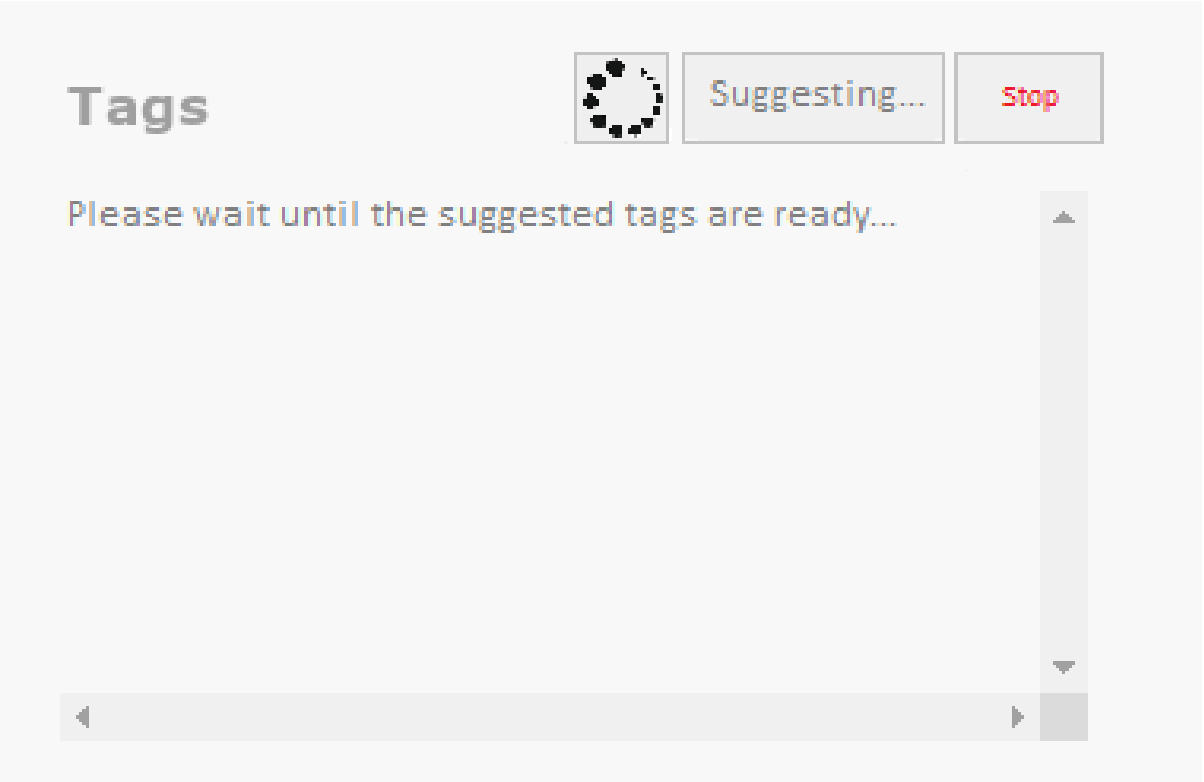

Figure 10.3: ARI tagging component during the suggestion process

After the tag suggestion process is finished the journalist is presented with a list of tag suggestions and the other tags as seen in figure 10.4.

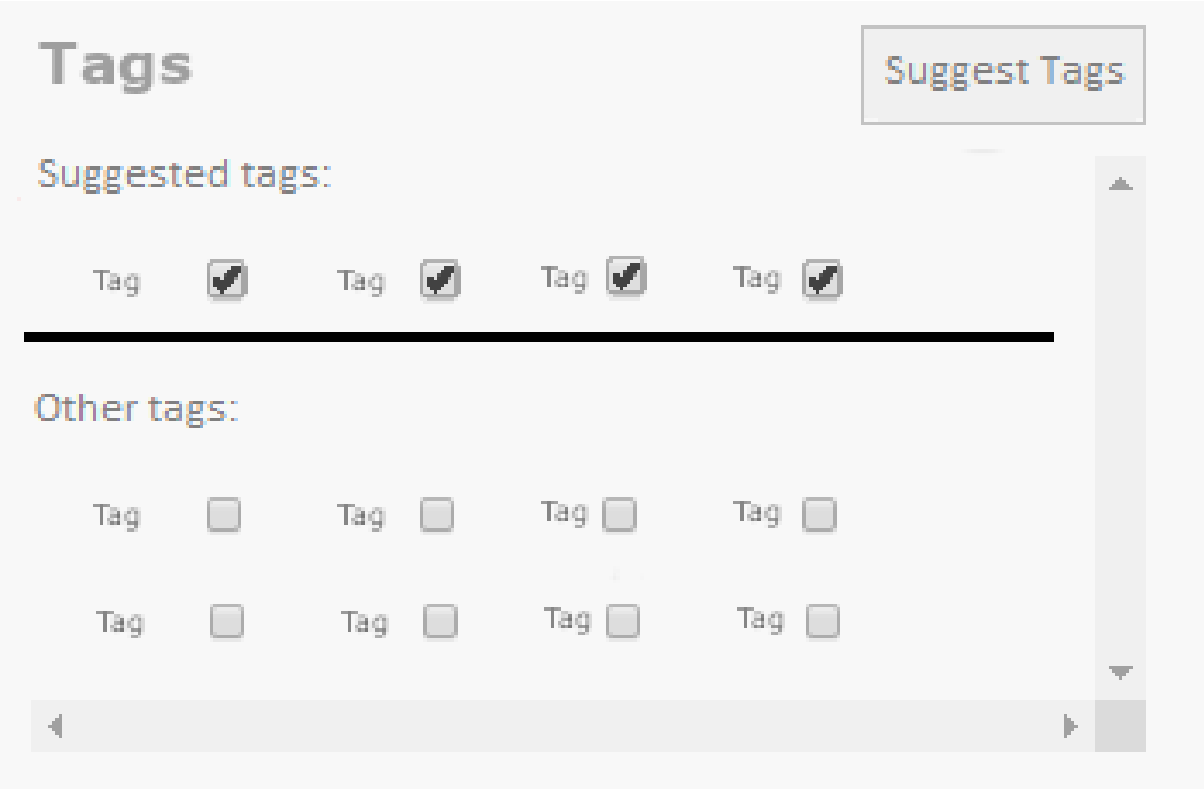

Figure 10.4: ARI tagging component after getting suggested tags

The journalist can mix as they like between suggested tags and other tags.

10.1.2 Interface evaluation

Heuristics	Description	Satisfied by
Effectiveness	User task completion with a high degree of accuracy	- Is a simple process with limited input and touch points therefore decreased chance of user error - Suggestion is an atomic operation, either it succeeds or does not - Simple language used
Efficiency	How fast can the user complete their task	- Tag suggestion takes one click - Tag selection is simple check-box click - Is an internal system thus communication time is shorter
Engagement	UX and aesthetics	- UI based on ARI and fits it design-wise - UI is in mock-up stage, therefore not finalized and easily replaceable thanks to separation of concerns
Error tolerance	Errors minimization and recovery	- In case of technical failure tag suggestion process can be repeated as many times as required - In case of irrelevant tag suggestions user can manually select tags
Ease of learning	How easy is it to learn the new process	- Users are journalists accustomed to similar process - Users understand concept of tagging and can use their existing mental models

Table 10.1: Overview of usability heuristics

10.1.3 Requirement Compliance

Req. ID	Req. Name	How they are met
FR-1	Tag suggestion and selection	Functional view: The tag functionality is added to the ARI by using the Tag Interface (design seen in section 10.1.1), Tag Mapper and Keyword Extractor. Information view: Tag suggestions with their relevance score are sent from the Tag Mapper to the ARI Tag Interface.
FR-2	Tag relevance	Functional view: The Keyword Extractor returns keywords with a relevance score that is used to suggest the most relevant tags. Information view: The relevance score is sent along with the suggested tags.
FR-3	Tag correction	Functional view: Manual selection and suggestion override are available in the ARI Tag Interface (as shown in Section 10.1.1 of Usability).
NR-1	Limited suggestions	Functional view: In the ARI Tag Interface only a limited number of tags will be suggested (as shown in section 10.1.1). This is done by ranking the tags based on relevance.
NR-2	Visibility	Functional view: The ARI Tag Interface notifies the user when the tagging process is in progress and when it is done. Section 10.1.1: Tags can be clearly seen along with progress indicator.

Table 10.2: Overview of usability requirement compliance

10.2 Regulation

Req. ID	Req. Name	How they are met
NR-3	Tagging standard	Context view: The tagging standard is stored in the Tag Mapper. Information view: The Tag Mapper has all IPTC subjects and the ARI Tag Interface is only aware of the IPTC root subjects (tags).

Table 10.3: Overview of regulation requirement compliance

10.3 Evolution

Due to Kristeligt Dagblad's expected change from Drupal 7 to Drupal 8, any new functionality has to be made to comply with Drupal 8 APIs.

When the tag suggestion system is implemented, there are required changes on multiple levels: the ARI, Drupal and the organization.

ARI changes

- Add a user interface for the tagging functionality
- ARI has to utilize the Tag Mapper, provide it correct input and handle its output

Drupal changes

- Database has to incorporate the Tag entity along with its relationship with Article

Organizational changes

- Users have to learn how to interact with the tagging features in the new system

10.3.1 Requirement Compliance

Req. ID	Req. Name	How they are met
AR-1	Portability	Context view: The Tag Mapper and Keyword Extractor are independent of ARI and thus not affected by changes to Drupal.
AR-2	Input & Output	Information view: The input and output of the Tag Mapper is clearly defined.
AR-3	Low-coupling and Integration	Context view: ARI and the Tag Mapper are coupled by the Tag interface. The Tag interface can be easily replaced without affecting the tagging solution.

Table 10.4: Overview of evolution requirement compliance

10.4 Performance and Scalability

Req. ID	Req. Name	How they are met
NR-4	Performance	To fully meet this requirement a development view should be made to describe how the system processes can be optimized. Based on simple performance test of a proof-of-concept implementation of the Keyword Extractor, we believe that this requirement is not compromised in the context of the proposed architecture.
NR-5	Concurrency	We believe that concurrency is not a substantial issue, as the tagging system is build only for Kristeligt Dagblad and has to handle only up to tens of concurrent users. Specific details related to concurrency should be presented in a concurrency view.
AR-4	Scalability	Since the number of users is limited, large spikes in usage are not expected. This requirement can be met similarly as NR-5.

Table 10.5: Overview of performance & scalability requirement compliance

10.5 Availability and Resilience

Req. ID	Req. Name	How they are met
AR-5	Error handling	Functional view: If an error occurs it is propagated back to the ARI Tag Interface where it is displayed to the user.
AR-6	Non-blocking behavior	Functional view: All other ARI functionality is available while the tag suggestion process is ongoing.

Table 10.6: Overview of availability & resilience requirement compliance

10.6 Localization

Kristeligt Dagblad publishes danish-only news media and therefore the tagging system nomenclature has to reflect that. As the IPTC NewsCodes Subjects are in English, an uniform translation of the Subject names is required from English to Danish. Ritzau, a danish news agency, provides such a translation and thus the system's tag terminology conforms to their standards.

Req. ID	Req. Name	How they are met
NR-6	Tag language	Information view: ARI and the Tag Mapper contain Danish Translations of the IPTC root subjects and use them as tags.

Table 10.7: Overview of localization requirement compliance

Part III

Appendices

Chapter 11

Appendices

11.1 Project Charter

11.1.1 Group Members

1. Daniel Șerbănescu
2. Frederik Zeilberger Thulstrup
3. Lukas Kucerik
4. Anton Kjær Hansen
5. Tomas Nemecek

11.1.2 Project Description

The purpose of this project is to provide an architectural analysis for Danish news media company, Kristeligt Dagblad, in regards to their enterprise and software systems, focusing on the editorial process. Based on this analysis, issues will be pointed out and architectural solutions presented.

11.1.3 Project Scope

The editorial process, which this project focuses on, encompasses multiple systems within the overall architecture and involves a multitude of stakeholders. These systems include, for example, the editorial subsystem, content database, content presentation layer and subsystem managing print media.

The areas out of scope for this projects are:

- subscription systems - systems handling subscriber information and payments
- analytics systems - systems tracking user behaviour and interaction
- marketing systems - systems handling advertisement and promotions
- user related systems - user authentication and self-service

Editorial Process

The editorial process is a chain of activities that lead from article creation to article publication and content sharing.

Written articles are put into the article creation system and published into the print version of the newspaper. Some of these articles are later imported into the public-facing website of the

company. This way, the public-facing website only contains some articles which helps to drive the sales for the print media.

There are also articles written exclusively for the website. This means that the publishing part of the current editorial process is unidirectional, print articles can be both published in physical media and uploaded to the website, while digital articles can only be uploaded to the website.

Audience

The audience for this report are IT professionals in Kristeligt Dagblad and Informatics teachers at Roskilde University.

Issue Prioritization and Scope Creep Management

In the beginning of the project, there were no obvious problems to solve. Problems crystallized during the stakeholder interviews and as-is analysis of the current architecture. Solutions for these problems were identified in the Identified Problematic Situations document, which can be found in section 11.6 in the Appendix.

Since this project did not have a clear objective in its inception, there needed to be a system in place to process, prioritize and possibly solve the found issues within the current architecture. As issues were discovered, they were collected in the aforementioned document and prioritized with the company's stakeholders. As a result of this prioritization we will be able to avoid scope creep, as only the chosen issues were focused on in this project.

11.1.4 Expected Returns

Kristeligt Dagblad can expect an external analysis of their current architecture as well as a proposal of possible architectural improvements. The analysis will be formed from considering the different viewpoints from various perspectives. The viewpoints and perspectives will be chosen based on relevance since not all view points and perspectives are applicable to the system.

The exact analysis will consist of a stakeholder analysis, different models of the as-is architecture and our proposed to-be architecture. The architecture will be visualized by making context, functional, informational models along with others which will be chosen at a later time.

11.1.5 Critical Success Factors / Key Performance Indicators

One of the main success factors is the degree of involvement of Kristeligt Dagblad. This project will require time investment from Kristeligt Dagblad as well as the access to their documentation. Time investment can be measured in terms of the frequency of meetings with the company itself, frequency of feedback exchange and stakeholder involvement in general.

The degree of documentation access can be measured in terms of how thorough and up-to-date the provided documents are. This might pose a minor obstacle due to the fact that the company is currently in the process of changing their software architecture (and thus some documentation might be slightly outdated in some cases).

Because of the process of changing architecture we will take a snapshot of the current architecture, which will be updated if there are some critical changes from Kristeligt Dagblad. We will therefore start from the same starting point considering the same concerns and limitations, however the final architectural design will most presumably differ.

The changing architecture at Kristeligt Dagblad encompasses moving from a multi-site architecture to a single site architecture. A multi-site architecture is using one instance of the CMS system[1] and serves multiple sites with similar content. With a single site there is a one-to-one relationship between the domain and the CMS system.

11.1.6 Assumptions and Prerequisites

The central assumption and prerequisite for this project to start is Kristeligt Dagblad's openness to share vital information with us. We also assume that both the company and project supervisors will provide enough feedback to revise the documents, for the architectural description to be valid and relevant.

11.1.7 Risks

There are several risks involved in this project. Most likely, the biggest risk is that we might not be able to get in touch with the stakeholders we need and we will not have enough time with each individual stakeholder. If that is the case we plan on creating questionnaires that will take only a few minutes for the stakeholders to fill out. This way we will be able to capture the largest amount of stakeholder concerns.

Another risk is that we cannot obtain the needed documentation to model the current architecture, but this risk is not too great since Daniel Șerbănescu has access to a lot of the documentation himself since he works in the company.

[1] Content management system for simplified website creation

ID	Risk	Affected Area	Probab. (0-100)	Impact (0-10)	Magnitude	Mitigation	Solution
1	Unavailable stakeholders	ALL	50	9	4.5	Try to talk to as many stakeholders as we can find.	Send short questions via email to relevant stakeholders.
2	Stakeholders wont be forthcoming with their answers	ALL	25	6	1.5	Speak simple language without specific terms.	Send the expected questions to stakeholders before interviews.
3	Stakeholder communication is limited by time constraints	ALL	70	5	3.5	Be as efficient as possible during our time with stakeholders.	Prepare thoroughly for each interview. Ask for more time.
4	It is not possible to get access to current documentation	ALL	10	9	0.9	Try to ask various stakeholders about documentation.	Ask many questions and derive the documentation from them.

Table 11.1: Risk Matrix

11.1.8 Organization, Roles and Responsibilities

It is worth to add additional emphasis on Daniel Șerbănescu in particular. This is due to the fact that Daniel Șerbănescu currently, and prior to project initiation, works as a Web Developer in the Digital Development Department of Kristeligt Dagblad. Daniel Șerbănescu therefore possesses unique insights into the organization as a whole as well as the specific department that will be of special interest to us making it only natural that Daniel Șerbănescu assumes the role of the Project Manger.

While Daniel holds many useful insights due to his position at Kristeligt Dagblad, he also possesses pre-conceived biases regarding the infrastructure and internal processes of Kristeligt Dagblad. We will keep this fact to mitigate his influence on the project's decisions.

11.1.9 Overall Plan

Architecture Status

Project owner or sponsor	Mads Emil Lidegaard initiated this project in order to have an overview over the IT architecture at Kristeligt Dagblad. He will use the architecture plans to make decisions about what should be implemented in the future and where to channel the resources.
Project Manager	Daniel Șerbănescu has to report progress to the project owner and facilitate meetings with stakeholders within the scope and purpose of the project.
Project Team	Daniel Șerbănescu Frederik Zeilberger Thulstrup Lukas Kucerik Anton Kjær Hansen Tomas Nemecek
Target audience	The stakeholders will provide inputs and feedback to the Project Team as needed and as the project proceeds.

Table 11.2: Roles and responsibilities

Kristeligt Dagblad is currently in the process of changing their IT and software architecture. This means that the outcome of this project might differ from or merge with the outcome of their architectural change. This project can be classified as an in-house project, due to the fact that it does not involve external stakeholders.

Project Status

Figure 11.1 presents the milestones of the project in relation to course deadlines. Note that from April the 4th until May the 9th, there will be feedback meetings held with both the company representatives and project supervisors which are not shown on the diagram for readability reasons. We will clarify the exact meeting days with the company and update the plan accordingly.

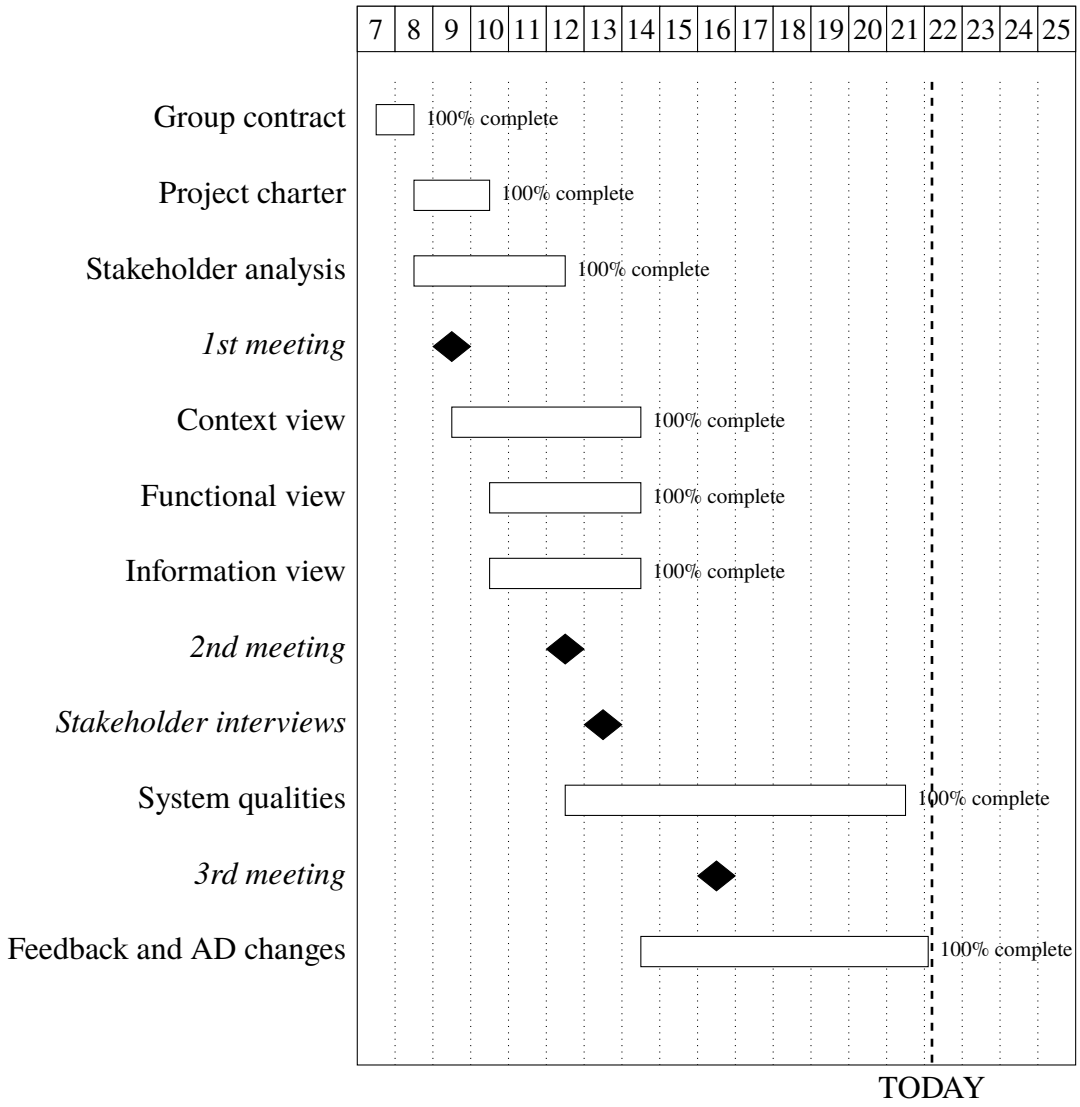

Figure 11.1: The progress of the course deliverables in relation to week numbers as Gantt Chart

11.1.10 Reporting and Evaluation

The deliverables will be in form of documents, charts and diagrams that are meant for documenting the systems used at Kristeligt Dagblad.

Each artifact will be peer-reviewed in class and feedback will be taken consideration. Afterwards the artifacts will be submitted to Kristeligt Dagblad for evaluation. This will be a formative evaluation providing us with feedback about how we are doing and what to improve. Since their time is valuable and they are a busy company we will strive for always providing high quality artifacts for them.

At the end of the project we will do a summative assessment of the whole project and specify how well our work has been able to help Kristeligt Dagblad with their future IT-architecture.

11.1.11 Communication Plan

The communication between the Group and Kristeligt Dagblad will be handled in the following ways:

- Face-to-face meetings
- Meetings through a delegate
- Written documents sent by email

Communication medium	Stakeholders involved	Frequency
Face-to-face meetings	Aquirers, Communicators	Once every 2-3 weeks
Meetings through a delegate	Communicators	Once a week
Written documents	Everyone, depending on which document is relevant for who	As often as needed

Table 11.3: Communication plans

Daniel Șerbănescu is responsible for the communication with KD. We will aim for a meeting with KD every 3 weeks on average.

11.2 Activities log

Date	Type	Agenda	Learning points
01 Feb.	Groupwork	Group formation	Got to meet each other.
05 Feb.	Groupwork	Project case initiation Initial work on contact letter for the case company.	We needed a tailor-made contact letter for the company. The template offered by RUC needed heavily modifications to fit our purpose.
07 Feb.	Meeting	Initial contact with Kristeligt Dagblad	Kristeligt Dagblad is on board for the project. It took some people skills to get the project approved.
11 Feb.	Groupwork	Set up team name Decide on group infrastructure, tools and workflow.	The 5 Horsemen was chosen as the team name. We kept as much as the old infrastructure we had in the 1st semester. And we decided to work with LaTeX as project writing tool, with git versioning.
14 Feb.	Groupwork	Initial work on the presentation for the first meeting with Kristeligt Dagblad Assign tasks and responsabilities.	We tried to get a focus on the full picture of what we are going to learn the current semester in order to present it at our first meeting with Kristeligt Dagblad.
21 Feb.	Groupwork	Initial work on the group contract and stakeholder interview questions. Deliver group contract. Make a timeline for the project with a Gantt Chart.	We struggled to get a complete picture over what it is we are going to learn and deliver to RUC and Kristeligt Dagblad. At the end we had a group contract and a draft of the questions for stakeholders.
25 Feb.	Groupwork	Preparation for the first meeting with Kristeligt Dagblad.	Got the final slides for presentation and prepared ourselves for the meeting we have the next day.
26 Feb.	Meeting	First meeting with Kristeligt Dagblad.	We got to see a functional diagram of the systems involved and gathered stakeholder concerns

26 Feb.	Groupwork	We work on: • Selected the editorial "data-flow" • Project charter v1.0 • Gantt chart with milestones • Stakeholder analysis v1.0	We took the feedback received at the first meeting and shaped our project path towards the editorial subsystem.
28 Feb.	Feedback session	Hand in project charter Receive feedback for it	There was uncertainty about: • Editorial process definition was renamed to content creation process and provided a better description • Deliverables were revised artifacts' description • Scope creep and how it is going to be handled • Which perspectives are going to be delivered. • Daniel Șerbănescu's role and his biases.
03 Mar.	Groupwork	We work on: • Overview over versioning workflow. • Review notes from 28th of February. • Implement feedback into project charter. • Pinpoint and limit the work area within Kristeligt Dagblad's systems. • Finish the questions for stakeholders.	We had collaborative issues with our main writing tool, Overleaf, so we keep an external backup of the project while continuing working with Overleaf.
05 Mar.	Groupwork	We work on: • Project charter terminology update (convert editorial to content creation). • Context view	There were no specific highlights for today.

07 Mar.	Feedback session	We received feedback on Stakeholder and Project Charter and generally about the document.	We learned that we have to be more specific and give examples about the problem at hand. We also need to be more consistent about terminology and document structure. We need to focus as well on specifying Kristeligt Dagblad's changing architecture and its effect.
12 Mar.	Groupwork	We sat up the Activities log and filled them in. Context view was heavily updated. Described the process and effect of Kristeligt Dagblad's changing architecture.	Changed from content creation to editorial. We need to use terminology that the client uses. Whenever we get into heated discussions we need to draw to get the point out faster. We delimited context view from functional view point as they are different facets of the editorial system. So we do not need to focus on only one specific viewpoint as the others might help with the establishment of the original viewpoint.
14 Mar.	Groupwork	Identified stakeholders we want to talk to and prepared questions for them. Updated Risk matrix Updated the Context View Initialized Functional View V1	We realized that we were missing the feeds in the context view and then revised the view to include it. Reinforced the lesson from last time: Drawing on whiteboard always helps.

17 Mar.	Groupwork	Initialized information view V1. Added taxonomy interface. Finalized functional view. Added highlights as type of data. Described editorial system in the context view.	We found out that we deal with a lot more data than we originally thought and we discovered new stakeholders. We also discovered two new systems that we should describe, i.e. Taxonomy Editor and Special Article Editor. We were able to identify two possible improvements for the Kristeligt Dagblad's architecture.
19 Mar.	Meeting		We learned that we need to adjust the functional diagram to make it reflect the real world. We learned from stakeholders about the adjustments we should make to the diagrams.
21 Mar.	Groupwork	Updated functional and Informational view diagrams and descriptions Selected a viewpoint and perspective to write about Finalized context view	Voting works
26 Mar.	Meetings	Met with with with the commercial director, digital secretary, Head of IT and Developer of Saxo databridge	We learned a lot about the system from the stakeholders we have talked to. We have also learned to be more prepared for the stakeholder meetings in the future - have more precise questions and have them in front of us while interviewing.
28 Mar.	Meetings	Met with with with the head of innovation and a print layout person	One of the lessons that was reinforced was the idea that stakeholders can have very different viewpoints on the processes and the issues within them.

31 Mar.	Groupwork	Added description of stakeholders in the stakeholder analysis. Made a list of issues for us to possibly solve. Redefined the concurrency view models.	There are no standard diagram that solves the issues of concurrency. It helps to listen to recordings when not recalling some parts of the interview.
02 Apr.	Groupwork	Cleaned up TODOs in our project Mentioned why ARI was created in the first place Updated gannt chart Fixed up concurrency view model and description Audience and status. Implemented AD template sections	Formatting helps making a point.
04 Apr.	Feedback session	We fixed small formatting issues with readability of the report.	We learned that we have to be more concise in writing.
09 Apr.	Groupwork	Implemented more of the AD template	We learned that we have to be more concise in writing.
16 Apr.	Meeting	Clearing up questions. Presenting and prioritizing problematic situations	We learned that presenting problematic situations, and letting the stakeholder prioritize them, worked well.
18 Apr.	Groupwork	Started on To-Be architecture	Having an actual problem description helps
20 Apr.	Groupwork	Worked more on context and information models	N/A
23 Apr.	Groupwork	Prepared AD for peer review	N/A
25 Apr.	Groupwork	Implemented review feedback, throwing Concurrency, started on requirements	Should have started on requirements and problem definition in general way earlier - would have helped us follow the AD template
27 Apr.	Groupwork	Reworked our functional diagram format into to UML activities	While our custom diagrams make sense to us - others have hard time following it. Makes sense to follow standards

4 May.	Groupwork	Cutting down on text	Our understanding of concise is different from others. We still write too much text even when we try be succinct.
6 May.	Groupwork	Cutting down on even more text	Same as on the 4th
7 May.	Groupwork	Taking Magnus' advice to heart and started front-loading the report	People do not want to read lengthy text and want to get to the point as fast as possible
9 May.	Groupwork	More front-loading	Same as on the 7th
11 May	Groupwork	Revised stakeholder analysis and concerns	We learned that we have to be more concise in writing.
14 May	Groupwork	Reflection workshop, creating learning points, fixing up AD	N/A
20 - 26 May	Groupwork	Writing reflection and trying to finish the AD and implementing Magnus' feedback	Having actual problem definition at the beggining would have so much time

Table 11.4: Table of activities log.

11.3 Stakeholder questions

- What systems would you consider to be part editorial system?
- who has access to journalist or other private information?
- Saving manhours by streamlining the editorial process to publish both on web and print?
- Why Kristeligt Dagblad have two departments for publishing (print and digital)?
- How complicated is the conversion from digital to print? How about print from digital?

Questions for meeting on the 18th of april:

- Is there a formal, or informal, process in place to train employees in using the systems encompassed in the editorial process?
- Does Kristeligt Dagblad-digital have a strategy for testing?
- Who actually finalizes a print article and sends it to print? Is tht an editor?
- Is there an estimated life expectancy of the ARI?
- Do you believe it is feasible to focus our work on implementing/"fixing" features in the ARI?

Questions for tagging solution

- Do you have a concrete list of tags which we should use or does it have to be written from scratch?
- How many tags would there be in such a closed list? Are talking about tens, hundreds or thousands?
- Should the tags be hierarchically ordered? This means that there would be parent tags which would have child tags - and perhaps multiple levels of these child tags.
- Relations between tags and categories

11.4 Stakeholder Analysis

Kristeligt Dagblad has been very helpful in regards to providing access to stakeholders. Majority of our meetings were with Mads Emil Lidegaard, the Head of Digital department, and Kristoffer Arrild, the software architect. In these meetings were figured out the scope of the project and what processes we should focus on. Afterwards, were also gained access to six other different stakeholder groups in Kristeligt Dagblad which take part in the editorial process.

11.5 Stakeholder Concerns

This section describes their main stakeholders concerns that we managed to gather over multiple meetings. More details and further stakeholder descriptions can be found in section 11.4 in the Appendices chapter.

Stakeholder	Concern
Head of Digital Department (Mads Emil Lidegaard)	Wants quick and continuous evolution of existing architecture as well as feasibility of the proposed architecture.
Software Architect (Kristoffer Arrild)	Data is a central to the company business and architecture should accommodate different platforms
Chief Revenue Officer (Hans Christian Kock)	Attracting and gaining new subscribers is paramount. Wants an improved tagging system for the journalists. Is unsatisfied with VisioLink
Digital Journalist (Julie Greve Bentsen)	Concerned with the editing of digital articles and specific issues and bugs in the ARI and related sub-systems.
Head of IT (Jørgen Schmidt)	In charge of the base IT-architecture and thinks ARI can be improved.
Former Developer Consultant (Carsten Dan)	Developed the Saxo-DataBridge, wants to improve it, as it is central to Kristeligt Dagblad.
Head of Innovation (Morten Thomsen Højsgaard)	In charge of Kristeligt Dagblad's digital content and believes an improved customer experience is essential for the company.
Page layout designer (Hannibal Erngaard)	Reliability and availability, as well as an improved UX, of the systems he uses on a daily basis.

Table 11.5: Stakeholder concerns

11.5.1 Stakeholder description and concerns

Mads Emil Lidegaard (Head of Digital Department)

- Thinks that quick evolution and continuous change is crucial for the architecture, since Kristeligt Dagblad is currently in the process of changing to a different architectural model.
- The people in the company are really busy and it is hard to make appointments with all relevant stakeholders. This might indicate heavy load on the operations and maintenance processes in the company which could be a good area for improvement.
- Based on our interview with him, his main concern was feasibility of the proposed architecture.

Kristoffer Arrild (Software Architect)

- Data is a central part of the system and it is crucial to the company business.
- The architecture should accommodate for different platforms (web, application) that access the content.
- Our primary contact.
- Is in charge of the architecture and his proposals have to be approved by the board of directors, which he is also a part of.

Hans Christian Kock (Chief Revenue Officer)

- Thinks that it is crucial for Kristeligt Dagblad to focus on making the subscription onboarding as easy as possible and the newspaper accessible throughout various platforms and devices.
- Print media is still very big part of the company's business, but the digital media has been growing rapidly in the last years.
- Kristeligt Dagblad strives to deliver the same product across different platforms. However there have been issues in regards to delivering the same experience on both iOS and Android. A lot of marketing, guides and user experience testing in regards to how to use Kristeligt Dagblad's mobile app has focused mostly on iOS. Recently however they have gotten complaints that the Android experience is different from iOS and is considered to be subpar compared to the iOS version. Kristeligt Dagblad hopes to make up for this oversight and thus increase their market penetration on Android devices.
- Kristeligt Dagblad main source of revenue are subscribers rather than advertisements.
- Hans would like to see a better solution for tagging articles since journalists tend to forget how many categories a certain article belongs to. Solving this issue would lead to a more accurate search functionality for articles based their topics. This, however, seems to be a broader issue as more media companies are facing the same problem.
- VisioLink does not seem to satisfy Hans' view of the company needs, since it tends to focus more on advertisement and he'd rather see more focus on subscribers. He suggested other alternatives like Twipe might make a better fit for Kristeligt Dagblad.

Julie Greve Bentsen (Digital Journalist)

- Julie as a Digital Journalist editing articles to be published on the website
- For Julie to start working on the articles in Drupal or ARI, they need to be imported from Saxo Databridge (this happens automatically). However, sometimes this import gets stuck and the articles do not go through - in this case, she is very confused about whether she should talk to Mads Emil Lidegaard (the problem is in Drupal) or to Jørgen (the problem is in Saxo or Saxo Databridge). She also mentioned that sometimes it is difficult to continue working when these problems arise during the night shift since there is no one to go to.
- Another issue that she runs into is improperly formatted articles in Saxo, which she cannot fix in ARI.
- When the articles are imported from Saxo, the changes from Saxo overwrite the changes in ARI. Julie mentioned that this occurs regularly and causes inconvenience as she either has to re-do her edits or wait until the end of the day to do them. Julie herself offered possible solution that she could mark some text as "not-updatable". Currently the journalists can now choose whether they want to update pictures, text, or none, but it would be useful if the text updates would be more granular.
- There seems to be a certain discomfort when using the systems among the journalists, since Julie mentioned that they are afraid to go off the regular way of doing things in the systems because they are unsure about what can go wrong when they do not do things the right way. The system does not inform the journalists about errors when importing or formatting articles, they just do not get imported when updates arrive. Julie would appriciate a better error information system, so that she knows where the issue lies and who she should turn to for help.
- Julie was working in Kristeligt Dagblad before the automatic import of articles was implemented (Saxo Databridge) and said that the workflow got significantly better because of that.
- Other issues with ARI:
  - Cannot select the main article picture from Drupal, they need to download it in SAXO and re-attach it.
  - There should be a default picture size automatically selected for article if the journalists do not do so.
  - Problem with italics when importing from SAXO to Drupal.
  - The problem when an article should go into tomorrow's issue in the print but should be published on the website today.
  - There is a lack of documentation for keyboard shortcuts in ARI

Jørgen Schmidt (Head of IT)

- Jørgen has been part of Kristeligt Dagblad for 40 years and started to work with SAXO in 1994. He has a very good understanding of the Saxo system and he was responsible for creating the base of IT infrastructure in the company.
- What is considered to be an issue is that there is no way to add related articles in SAXO, it can only be done through ARI.
- He suggested that it might be better if the articles were only written in ARI/Drupal and then transferred to the Saxo system. However when this was brought up with Mads Emil Lidegaard he considered it a non-issue, as simply copying the article takes minimal effort.
- There are different versions of articles in Drupal and Saxo, this could somehow be improved - this points to possible article synchronization issues that Julie Greve Bentsen mentioned as well.

Carsten Dan (Former Developer Consultant at Kristeligt Dagblad)

- Carsten helped to develop the Saxo Databridge as solution for synchronization issues between Saxo and Drupal a couple of years ago.
- He seemed to pinpoint the cause of the article synchronization problem to be the half-an-hour synchronization delay between Saxo and Saxo Databridge
- Carsten sees Saxo as a very old system that is evolving very slowly because of its superiority on the market which makes Saxo's developer-facing side years-old and very difficult to work with. That is why the Saxo Databridge was developed - to reformat the articles from Saxo format into a format that Kristeligt Dagblad's systems can work with more easily.
- Even though he does not seem to like Saxo and the way it integrates with other systems, he believes that Kristeligt Dagblad is on the right track when using Saxo Databridge correctly and that they should keep on improving it.

Morten Thomsen Højsgaard (Head of Innovation)

- Morten Thomsen Højsgaard is in charge of all digital content and is responsible for about 30 people.
- He is very satisfied with the current system in regards to the editorial process, especially compared to the previous systems before 2017 and the systems that other companies are using.
- While Morten Thomsen Højsgaard can understand some issues regarding the search for related articles and people having to wait until the end of the day to finalize the article, however the effort he perceives would have to go into fixing their internal processes would, according to him, be better spent on improving the customer experience.

Hannibal (Page layout designer)

- Hannibal is in charge in of designing the layout for individual newspaper pages.
- For his daily tasks he uses the InDesign program which is connected to the Saxo system and makes sure that the articles and images fit correctly on the page.
- His main concern is the simplicity of the UX experience, he feels that to achieve his main task he has to click too much and goes through too many system windows.
- Another of his larger concerns is the reliability of the system itself. Every minute lost in the editorial process can result in serious problems down the line when the articles need to be sent to the printing presses.

11.6 Problematic situations

1. Article sync
Within the editorial process there is a bottleneck in regards to editing articles. To edit an article in ARI, to be able to receive updates from Saxo one has to have updates enabled which can overwrite the edits done in ARI. This setting is enabled by default, other settings are to receive only updates for text or no updates at all. In practice that means that editors for digital articles have to wait until the end of the day for the print journalists to finish updating their articles, and then can they edit the article for the digital version.

2. Tagging of articles in ARI
When journalists write their articles it is up to their discretion to tag the articles. This means that there is a difference between how and how much each article is tagged, because it is up the journalists themselves. This leads to unreliable tagging of article which means that subscribers cannot find relevant articles under a specific topic.

3. Search for relevant articles
Journalist have to add related articles to the article that they are writing. This means that they have to search the database based on some keywords. However currently the search engine searches article on the exact match. This means that unless the journalist can find the exact wording of the article title, they will not be able to find the article.

4. Print article collection on web
Currently there is no functionality on the Kristeligt Dagblad's domain that naturally leads from one article to another similar to the way it works in print newspaper or in Visiolink. The database already contains all the data regarding the publication date and the newspaper page number of the article.

5. Main article picture is not easily selected
Currently the digital editor cannot easily change the main picture of an article.

6. No ARI keyboard shortcut documentation
Digital journalists do not know the shortcut for efficiently operate ARI. They would like to have some documentation.

7. No default main article picture size
When articles are imported from Saxo, the main article picture does not have a default size, meaning that the journalist have to manually select the size even-though the size should be generally the same.

8. Better alert system
While the developers do have an alert system, there is no sticky functionality, meaning if for example if a system goes down after 4pm the notification can be pushed out by other notifications until the next day. That means that no one might notice for a while that a system is actually down. Access to the alert system could then also be extended to other employees, meaning that if they run into an issue they can see where the issue might be and know who to contact about it.

9. Italics export incorrectly into ARI
Italic text is not correctly exported from Saxo to ARI

10. No password policy
Currently there is no password policy on the website or in the Drupal system itself. This means

that in theory a user can have 1 character password. These users can then be promoted to admins, which means that admin account are vulnerable to password guessing and brute force attacks.

11. There is no way to get from audio article to the text version
Currently there is no way to find the text version of an audio article. While there is a link from the text article to the audio article, there is none the other way.
The title of the audio version is also different from the text version, meaning that the user cannot search the site by the title.

ID	Description	Frequency	Stakeholder relevance
1.	Article sync	Once a week	Low
2.	Tagging of articles	Always	Highest
3.	Search for relevant articles	Always	Medium
4.	Articles from current issues.	Always	Lowest
5.	Main picture of article	1 out of 2 days	Low
6.	Keyboard shortcuts documentation for ARI	Always	Medium
7.	Default picture size when importing from ARI	Always	Medium
8.	Better alert system for journalists	Always	High
9.	Italics problem in ARI	Always	Medium
10.	Password policy for users	Always	Low
11.	Link audio articles back to digital article	Always	Lowest

Table 11.6: Table of identified problems with their frequency.

11.7 Information view entities

Author
Author is another of the main entities, it represents who participated in writing the article. Articles can therefore have multiple authors.

Audio Article
Kristeligt Dagblad uses the Nota's service to turn selected articles into audio format. Thus some of the articles have a audio file attached to them.

Audio issue
Audio articles are periodically collected into "issues", which are then published on Kristeligt Dagblad's website.

Special article
Some of the articles are so called "Special stories" which are articles that do not follow the normal article template. These article can contain multimedia or interactive elements. These articles are not written in the usual way, an HTML file is created which is then connected to an article, the bearing article.

Topic Highlight
Article topic highlight is a collection of articles based on a specific topic. These highlights are curated and manually kept up to date.

Frontpage Highlight
Frontpage highlight is a collection of manually selected articles that appear on the frontpages of Kristeligt Dagblad's domains.

Image
Articles usually contain images. An image can be present in multiple articles.

Video
Articles can contain videos as well as images, these videos can also be in multiple articles.

Element
An element is a part of an article and it is used within the body text of an article. It can be an image, a video, a link to a related article etc..

11.8 The security perspective

There are a number of security issues in relation to the editorial system. Security is not really part of our scope, but we have identified some security threats that could be useful to Kristeligt Dagblad to consider. By looking at the confidentiality, integrity, availability and accountability we can categorize the threats.

11.8.1 Confidentiality & Integrity

In the editorial system there is not that much sensitive information. It is mostly that the paid articles should be hidden for non-subscribers and unpublished articles should be available only to journalists. Therefore, there are multiple threat vectors:

- Someone may obtain unauthorized access to unpublished articles
- A non-paying subscriber may find a a way to access paid articles
- Physical manipulation with servers may compromise confidentiality

We do not consider these threats as serious, as the current system to take care of these threats. We do not see any reason to worry about this issue.

A possible threat to Kristeligt Dagblad could be that articles are getting modified without anyone knowing how and by whom.

Solution: Kristeligt Dagblad already has a fairly robust article versioning system in place which takes care of this issue.

Two potential security issues, which overlaps both confidentiality and integrity, concerns potential unauthorized access to the ARI.

1. As there is no password policy in place, chances are that some users may use weak passwords.
2. Multi-factor authentication (MFA) is not supported for the ARI.

Solution:
1. Implement a password policy for the ARI.
2. Implement MFA for the ARI.
3. Consider to enforce use of a virtual private network (VPN) and only allow traffic to and from the ARI from within the private network.

11.8.2 Availability

Articles' availability is vital for Kristeligt Dagblad. Any server unavailability would cause great harm.

- DDoS attack causes servers to go down and therefore making everyone unable to access articles
- Kristeligt Dagblad's servers may be struck with malware.

Solution:

- Kristeligt Dagblad has a mitigation strategy for this threat by hosting their systems on third party servers with a backup policy and redundancy in place.
- DDoS-protection

11.8.3 Accountability

Kristeligt Dagblad has a system in place to track changes to articles. This provides an overview of revisions to articles and thereby accountability for journalists.

Furthermore, Kristeligt Dagblad has a syslog-server, with an ELK-stack on top, serving as an informal SIEM.

11.8.4 Security Analysis of Context view

To analyze security risks in relation to the editorial process we have used the CIA goals [RW12, p.444] to categorize the different risks.

Confidentiality

There are both premium and unpublished articles the company wants to hide from certain users of the system. There are several ways confidentiality could be misused:

- Someone without administrator access gets access to unpublished articles
- Someone who has no subscription finds a way to access the premium articles
- Someone without administrator access gets access to the journalists personal information such as name, email, image, birth date and so on.

Integrity

There are several ways in which the integrity of data can be jeopardized:

- Articles and author data being changed without authorization. This could happen if someone gets access directly to the database.
- Someone manages to intercept the article data when it is being sent to the database. Then they can edit it and resend it again. This way articles could be modified to something else than the journalist expected.
- A vulnerability in the Drupal installation causes articles to become modified by an unauthorized person.

Availability

Data should be available to the users with permission at all times. There are several ways availability could become obstructed:

Other security risks

We identified security risks at this level to be the link between Editorial System and Internal Storage DB and Editorial System to ActiveCampaign. These seem to be most vulnerable links between the systems where the modified or unavailable article data can have the most impact on the readers or journalists.

11.9 Connected Entities

Internal Entities

SAXO Databridge	As part of the editorial process, journalists can input their articles into the SAXO system which is an external system independent of Kristeligt Dagblad. The SAXO databridge acts as an adapter between the SAXO system's article format and the editorial system's article format.
KD domains	Under KD domains we understand all the different websites that Kristeligt Dagblad publishes their articles to, they are listed as follows: www.kristeligt-dagblad.dk, forlag.kristeligt-dagblad.dk, www.religion.dk, www.kristendom.dk, www.etik.dk, www.pilgrimsvandring.dk.

External Entities

Nota	An external system that converts the text of the articles into an audio format.
Ritzau	Provides overview of top news stories for the danish market. The editorial system receives selected articles from Ritzau. The article selection happens automatically based on the article's category.
ActiveCampaign	A system for managing newsletter subscriptions. Selected articles based on topics that the subscriber signed up for are sent through ActiveCampaign via email.
Feeds	Editorial system is connected with multiple feeds which it exports the articles to. These feeds include Facebook Instant Articles, RSS, Ni.dk and Churchdesk.

Data Store

Internal storage database stores all the website data, including the articles.

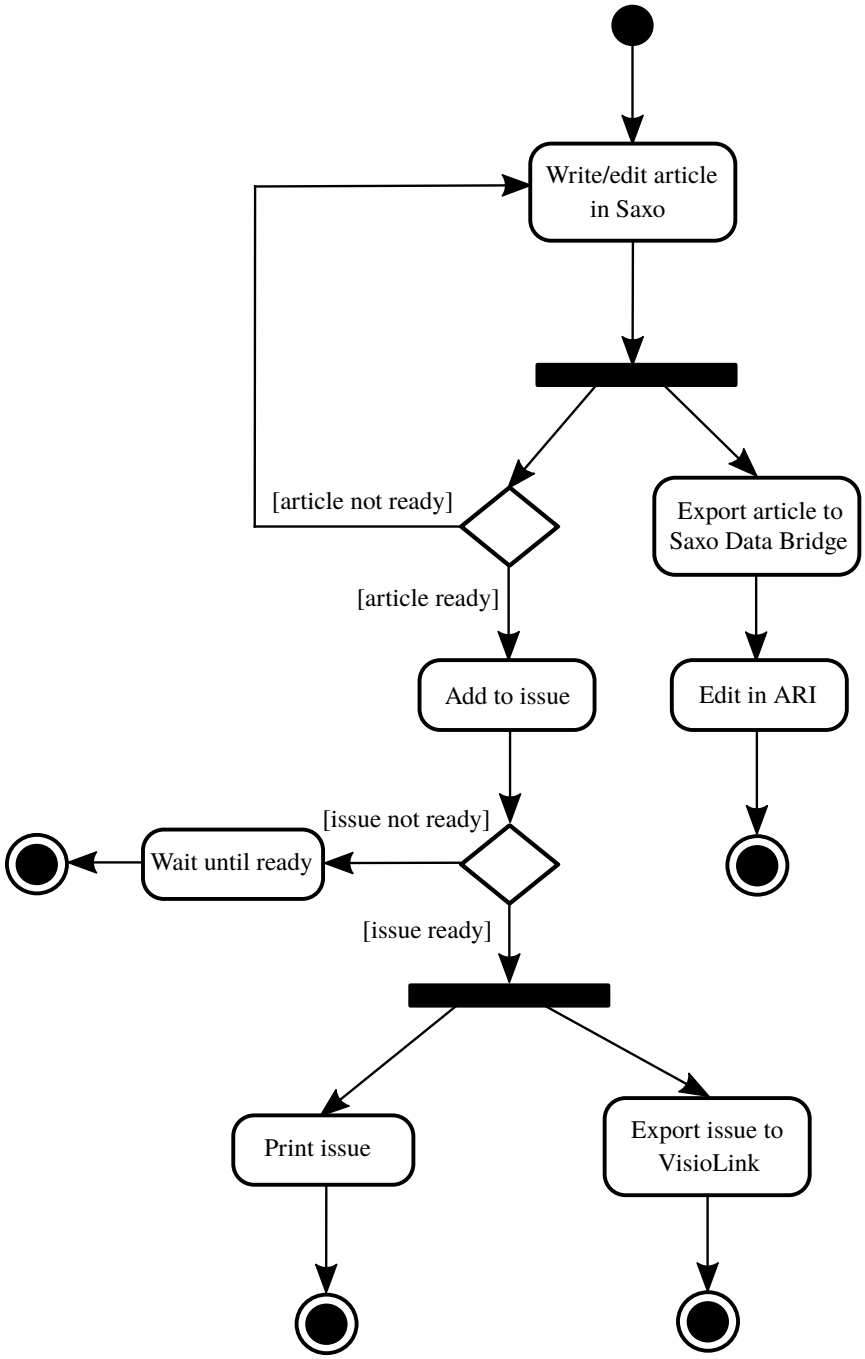

Figure 11.2: UML activity diagram showing the Saxo-related section of the editorial process

One of the ways that the editorial process starts is in **the Saxo system**, which is an external entity to Kristeligt Dagblad. Articles written and edited in this system.

Each update in Saxo is sent downstream to **the Saxo Databridge**. That converts the Saxo-specific XML format into a format that is compliant with **the ARI system** article schema. This allows articles from Saxo System as well as articles inputted straight into the ARI to be saved

to the database. The ARI system is explained further in figure 8.2.

When the article is considered finished in Saxo, it is added to an issue. When the issue has all the articles, it is ready for print. It then sent to printing presses and exported to **Visiolink**. Visiolink is another external entity to Kristeligt Dagblad. It converts the collection of articles in a newspaper issue into a PDF. Kristeligt Dagblad's websites then use the Visiolink app to serve the PDFs to allow subscribers to list through a PDF of an newspaper issue.

11.10 IPTC Root Tags

ID	English Name	Danish Name - (Explanation)
01000000	Arts, Culture and Entertainment	KUL-(Kunst/Kultur/Fornøjelse)
02000000	Crime, Law and Justice	KRI-(Kriminal/Lov/Ret)
03000000	Disaster and Accidents	ULY-(Katastrofer/Ulykker)
04000000	Economy, Business and Finance	ØKO-(Økonomi/Erhverv/Finans)
05000000	Education	UDD-(Uddannelse)
06000000	Environmental Issue	MLJ-(Miljø/Klima/Natur)
07000000	Health	SUN-(Sundhed)
08000000	Human Interest	FOF-(Folk og Fæ/Kuriositeter)
09000000	Labour	ARB-(Arbejdsmarked/Pensioner)
10000000	Lifestyle and Leisure	FRI-(Livsstil/Fritid)
11000000	Politics	POL-(Politik)
12000000	Religion and Belief	REL-(Religion/Kirke)
13000000	Science and Technology	TEK-(Teknik/Forskning/Videnskab)
14000000	Social Issue	SAM-(Samfund/Socialt/Mennesker)
15000000	Sport	SPO-(Sport)
16000000	Unrest, Conflicts and War	URO-(Krig/Konflikt/Uro)
17000000	Weather	MET-(Vejr)

Table 11.7: IPTC Root Tags with Danish Translation

Part IV

Bibliography

Bibliography

[AL12] Scott W. Ambler and Mark Lines. *Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise*. Upper Saddle River, NJ: IBM Press, 2012. 513 pp. ISBN: 978-0-13-281013-5.

[AMa] Agile Modeling (AM). *Big Modeling Up Front (BMUF) Anti-Pattern*. accessed on 2019-05-26. URL: http://www.agilemodeling.com/essays/bmuf.htm (visited on 05/26/2019).

[AMb] Agile Modeling (AM). *Model Storming: An Agile Core Practice*. Last accessed on 26 May, 2019. URL: http://agilemodeling.com/essays/modelStorming.htm (visited on 05/26/2019).

[EU] The European Union (EU). *EUGDPR – Information Portal*. Last accessed on May 26, 2019. URL: https://eugdpr.org/ (visited on 05/26/2019).

[Her+12] Morten Hertzum et al. "Pilot Implementation: Learning from Field Tests in IS Development". In: *Communications of the Association for Information Systems* 30 (2012). ISSN: 15293181. DOI: 10.17705/1CAIS.03020. URL: https://aisel.aisnet.org/cais/vol30/iss1/20 (visited on 05/26/2019).

[Jyt16] Brender McNair Jytte. "Theoretical Basis of Health IT Evaluation". In: *Studies in Health Technology and Informatics* (2016), pp. 39–52. ISSN: 0926-9630. DOI: 10.3233/978-1-61499-635-4-39. URL: http://www.medra.org/servlet/aliasResolver?alias=iospressISBN&isbn=978-1-61499-634-7&spage=39&doi=10.3233/978-1-61499-635-4-39 (visited on 05/26/2019).

[Kom] Andreas Komninos. *An Introduction to Usability*. The Interaction Design Foundation. Last accessed on 26 May, 2019. URL: https://www.interaction-design.org/literature/article/an-introduction-to-usability (visited on 05/26/2019).

[LBP12] Chris Leadbetter, Roger Blackford, and Tony Piper. *Cambridge International AS and A Level Computing Coursebook*. Cambridge: CAMBRIDGE UNIVERSITY PRESS, 2012. 438 pp. ISBN: 978-0-521-18662-9.

[Mar04] M Lynne Markus. "Technochange Management: Using IT to Drive Organizational Change". In: *Journal of Information Technology* 19.1 (Mar. 2004), pp. 4–20. ISSN: 0268-3962, 1466-4437. DOI: 10.1057/palgrave.jit.2000002. URL: http://journals.sagepub.com/doi/10.1057/palgrave.jit.2000002 (visited on 05/26/2019).

[ML] MonkeyLearn (ML). *Topic Analysis: A Comprehensive Guide to Detecting Topics from Text*. MonkeyLearn. URL: https://monkeylearn.com/topic-analysis (visited on 05/26/2019).

[MT04] Rada Mihalcea and Paul Tarau. "TextRank: Bringing Order into Text". In: *Proceedings of EMNLP 2004*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 404–411. URL: https://www.aclweb.org/anthology/W04-3252.

[MT16] Malcolm Masso and Cristina Thompson. "Attributes of innovations and approaches to scalability – lessons from a national program to extend the scope of practice of health professionals". In: *Journal of Multidisciplinary Healthcare* Volume 9 (Aug. 2016), pp. 401–410. ISSN: 1178-2390. DOI: 10.2147/JMDH.S111688. URL: https://www.dovepress.com/attributes-of-innovations-and-approaches-to-scalability-ndash-lessons--peer-reviewed-article-JMDH (visited on 05/26/2019).

[MW11] Benjamin Mwanzia Mulili and Peter Wong. "Continuous organizational development (COD)". In: *Industrial and Commercial Training* 43.6 (Sept. 6, 2011), pp. 377–384. ISSN: 0019-7858. DOI: 10.1108/00197851111160513. URL: https://www.emeraldinsight.com/doi/10.1108/00197851111160513 (visited on 05/26/2019).

[OU] The Open University (OU). *Invention and innovation: An introduction*. OpenLearn. URL: https://www.open.edu/openlearn/science-maths-technology/engineering-and-technology/design-and-innovation/invention-and-innovation-introduction/content-section-0 (visited on 05/26/2019).

[PB08] Jan Pries-Heje and Richard Baskerville. "The Design Theory Nexus". In: *MIS Q.* 32.4 (Dec. 2008), pp. 731–755. ISSN: 0276-7783. URL: http://dl.acm.org/citation.cfm?id=2017399.2017405.

[Rog83] Everett M. Rogers. *Diffusion of innovations*. 3rd ed. New York : London: Free Press ; Collier Macmillan, 1983. 453 pp. ISBN: 978-0-02-926650-2.

[RW12] Nick Rozanski and Eoin Woods. *Software systems architecture: working with stakeholders using viewpoints and perspectives*. 2nd ed. OCLC: ocn755213567. Upper Saddle River, NJ: Addison-Wesley, 2012. 678 pp. ISBN: 978-0-321-71833-4.

[SB18] William Stallings and Lawrie Brown. *Computer security: principles and practice*. Fourth Edition, global edition. OCLC: 1024109104. New York, NY: Pearson, 2018. 800 pp. ISBN: 978-1-292-22061-1 978-0-13-479410-5.

[SR13] Jesper Simonsen and Toni Robertson, eds. *Routledge international handbook of participatory design*. New York: Routledge, 2013. 294 pp. ISBN: 978-0-415-69440-7 978-0-203-10854-3 978-1-136-26619-5.

[Vit19] Vitaliy. *Multilingual Rapid Automatic Keyword Extraction (RAKE) for Python: vgrabovets multi_rake*. original-date: 2018-06-11T12:19:07Z. May 11, 2019. URL: https://github.com/vgrabovets/multi_rake (visited on 05/26/2019).

[Wel13] Gordon Welty. *Strategy and Tactics for Pilot Implementation in the ADDIE Model*. May 30, 2013. URL: http://www.ivtnetwork.com/article/strategy-and-tactics-pilot-implementation-addie-model (visited on 05/26/2019).

[WIK19a] (WIKI). *Iterative and incremental development*. In: *Wikipedia*. May 20, 2019. URL: https://en.wikipedia.org/w/index.php?title=Iterative_and_incremental_development&oldid=898020438 (visited on 05/26/2019).

[WIK19b] (WIKI). *tf–idf*. In: *Wikipedia*. Page Version ID: 898834016. May 26, 2019. URL: https://en.wikipedia.org/w/index.php?title=Tf-idf (visited on 05/26/2019).