Knowledge sharing platform.

Module Code – COMP37006

Module Title – Workplace project

Module Leader – Dr Beverley Cook

Word Count – 0

Content Page Count – 0 *(excluding contents, list of tables, references, and annex)*

Contents

[List of Tables 3](#_Toc126091168)

[1. Executive summary 4](#_Toc126091169)

[1.1 Introduction 4](#_Toc126091170)

[1.2 Opportunity 5](#_Toc126091171)

[1.3 Gap Analysis 5](#_Toc126091172)

[1.4 Feasibility Analysis 5](#_Toc126091173)

[1.4.1 Operational 5](#_Toc126091174)

[1.4.2 Technical 5](#_Toc126091175)

[1.4.3 Schedule 6](#_Toc126091176)

[1.4.4 Economic 6](#_Toc126091177)

[1.4 Risk Analysis 6](#_Toc126091178)

[2. Technical literature review 7](#_Toc126091179)

[2.1 Team on MS Teams 7](#_Toc126091180)

[2.1.1 Limitations 7](#_Toc126091181)

[2.2 Dedicated SharePoint site 7](#_Toc126091182)

[2.2.1 Limitations 7](#_Toc126091183)

[2.3 Miro board 8](#_Toc126091184)

[2.3.1 Limitations 8](#_Toc126091185)

[2.4 Nuclino 8](#_Toc126091186)

[2.4.1 Limitations 9](#_Toc126091187)

[2.5 Comparison of technologies 9](#_Toc126091188)

[2.6 Summary 9](#_Toc126091189)

[2.6.1 Challenges 9](#_Toc126091190)

[3. Project management 10](#_Toc126091191)

[3.1 Scope 10](#_Toc126091192)

[3.2 Schedule plan 10](#_Toc126091193)

[3.3 Resource planning 10](#_Toc126091194)

[3.4 Gantt Chart & PERT Diagram 11](#_Toc126091195)

[3.5 Analysis of Legal, Social, Ethical and Professional Issues 11](#_Toc126091196)

[3.4.1 Legal Issues 11](#_Toc126091197)

[3.4.2 Social Issues 12](#_Toc126091198)

[3.4.3 Ethical Issues 12](#_Toc126091199)

[3.4.4 Professional Issues 13](#_Toc126091200)

[3.5 Risk Assessment 13](#_Toc126091201)

[4. Requirement analysis 14](#_Toc126091202)

[4.1 Research and Discovery 14](#_Toc126091203)

[4.2 Software Development Methodologies 14](#_Toc126091204)

[4.2.1 Waterfall 14](#_Toc126091205)

[4.2.2 Agile/Scrum 15](#_Toc126091206)

[4.2.3 Chosen Development Method 15](#_Toc126091207)

[4.2 Functional and Non-Functional requirements 16](#_Toc126091208)

[4.3 Use Case Analysis 17](#_Toc126091209)

[4.3.1 Use Case #1 17](#_Toc126091210)

[4.3.2 Use Case #2 17](#_Toc126091211)

[4.3.3 Use Case #3 18](#_Toc126091212)

[4.4 MoSCoW 19](#_Toc126091213)

[4.5 User stories 20](#_Toc126091214)

[5. Design 21](#_Toc126091215)

[6. Implementation 21](#_Toc126091216)

[6.1 Coding standards 21](#_Toc126091217)

[6.1 SOLID 21](#_Toc126091218)

[6.2 Design patterns 21](#_Toc126091219)

[7. Results 22](#_Toc126091220)

[8. Conclusion 22](#_Toc126091221)

[9. References 23](#_Toc126091222)

[10. Annex 23](#_Toc126091223)

[6.1 Annex 1 23](#_Toc126091224)

# List of Tables

[Table 1 - Gap Analysis 4](#_Toc119018113)

[Table 2 - Risk analysis 5](#_Toc119018114)

[Table 3 - Risk Assessment 12](#_Toc119018115)

[Table 4 - Functional and non-functional requirements 15](#_Toc119018116)

[Table 5 - Use case 1 16](#_Toc119018117)

[Table 6 – Use case 2 17](#_Toc119018118)

[Table 7 - Use case 3 17](#_Toc119018119)

# Executive summary

## 1.1 Introduction

A software system known as a knowledge sharing platform has both contemporary features that enable users to communicate information in real time and as a knowledge base. Users from around the organisation can post content or pose questions using this technology and, subject matter experts, can publish their comments in response.

Items are often shared between departments on an ad-hoc, informal basis, which can lead to duplication of effort or (unintended) exclusion of departments from useful findings.

Departments include:

* Product Design – who need to understand the needs and challenges of customers, what competitors offer, market trends etc through means such as desk research, webinars, conferences, and customer events/feedback.
* UX – who need to understand how to present the requirements of product features in the UI, using desk research, user testing and customer events/feedback.
* Research – who look towards the horizon of new technology and its applications, through desk research, webinars, conferences, early access demos, in-house trials, and developing prototypes.
* Marketing – who need to understand MHR’s place in the market, the needs and challenges of potential customers and the competitor space, though means such as desk research, webinars, conferences, and customer events/feedback, as well as access to research institutions such as Gartner and IDC.

Other departments around the company will also do their own research, such as DPO for keeping up to date on data privacy regulations and their effect on product development, but for the scope here, we will focus on the above four departments.

Items discovered/produced because of the research take on different forms, such as:

* Links (to articles, solutions, competitor websites, upcoming webinars, etc)
* Papers (academic papers, whitepapers produced by research organisations and/or competitors, etc)
* Books (several in the Research bookcase, and more dotted around the company)
* Videos (webinar recordings, YouTube videos, Teams meeting recordings, etc)
* Summary write ups/word documents/blogs/presentations produced (e.g. as a result of an investigation or of the findings from a conference)
* Excel spreadsheets of analysis (e.g. competitor analysis)

Items are shared between departments on an ad-hoc and informal basis. This sometimes leads to duplication of effort or (unintended) exclusion of departments from useful findings.

Means of ad-hoc sharing currently include:

* Email and Teams messaging
* SharePoint and OneDrive
* People First communities
* Miro
* Ad-hoc conversations

## 1.2 Opportunity

MHR is a leading HR service provider who strives for continuous evolution of its product, which requires mass research be shared across the business. Currently departments are siloed between each other, causing harm to company culture, and decreasing efficiency. The opportunity to bring in a revolutionary way to share knowledge between departments would be a huge asset to the business. Initial improvements would be seen in the reduced duplication of effort, and as time advances it would encourage the reduction of silos by increased collaboration, sharing of knowledge, and between department idea generation.

## 1.3 Gap Analysis

Table 1 shows a gap analysis which identifies and examines two primary areas that this proposed initiative addresses: two business procedures requiring development.

We are investigating a new method for between department idea generation and sharing of such ideas and knowledge, whilst allowing users to search and comment on posted content.

Table 1 - Gap Analysis

|  |  |  |
| --- | --- | --- |
| Current State | Desired State | Action to state |
| Content is currently shared between departments on an ad-hoc, informal basis, which can lead to duplication of effort or (unintended) exclusion of departments from useful findings. | An application for departments to manage and tag content. | Develop a cloud application where departments can share and tag content for other departments in a social feed. |
| Presently there is no access to globally search across the business for knowledge content, which causes increase of time due to not knowing content location. | Simplistic searching of content to allow users to use subjects. | Tagging of content and provide search functionally to filter down content. |
| Content has no managed structure and data is shared in different formats, making it difficult to analyse and report on. | Provide basic content structure and types to allow for manageable knowledge resources. | The system should allow users to enter content via forms which has strict content types, links, word documents, images, etc. |
| Content shared around the business has no feedback feed for employees to share thoughts on the knowledge resources. | Have collaborative feedback feature to allow for employees to feedback. | Have comments to be posted on the content for the author to review and reply. |

## 1.4 Feasibility Analysis

### 1.4.1 Operational

Departments will need to ensure that their staff will use this system in part of the process of knowledge sharing. It will become integrated into day-to-day business operations of sharing information. This system will be trialled on selected departments before being rolled out to the rest of the business.

### 1.4.2 Technical

This project will use the identical technology stack as other products within the business, meaning there will not be a need for project participants to receive additional training. The stack is Microsoft.NET for backend services, Angular for front end application and SQL database for persistence layer. The code will be deployed and maintained on the same cloud platform, Azure.

### 1.4.3 Schedule

A single team could take on this project, or run it in parallel with a current project, as there are several development teams operating within the company, provided they have redundancy and multiple developers per discipline.

### 1.4.4 Economic

As this is intended to be an internal tool, rather than a sellable product or to incorporate into another product, the project’s return on investments will be modest. Although it will not generate sales or outside income, it reduces duplication of effort around the departments and increase idea generation between departments.

## 1.4 Risk Analysis

Table 2 shows the risk analysis for the system, identifying risk itself, the probability, impact, and how we can mitigate or reduce the risk.

Occurrence probability [1 unlikely – 5 very likely]

Impact [1 low – 5 very high]

Table 2 - Risk analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #ID | Risk | Likelihood | Impact | Mitigation |
| 1 | Requirements are not fit | 3 | 5 – Requirements gathered does not match what the stakeholders want out of the system. | Complete business cases and provide questions for project stakeholders and document everything. |
| 2 | Scope Creep | 5 | 5 - When scope creep results in software development projects missing deadlines and taking longer than expected, risks can result. | Get the management approval before putting the project scope down in a Project Initiation Document. Throughout the project, refer to it and compare all modifications to it to ensure that they are in line with the Business Case. |
| 3 | Poor risk management | 3 | 4 – Poor risk management can lead to disruptions. These setbacks frequently result in one or more serious failures of a poorly thought-out software development project. | Create an in-depth risk assessment for the project which covers all risk and its mitigations. |

# Technical literature review

Below are existing systems/tools that were considered, and their suitability compared with the gap analysis.

## 2.1 Team on MS Teams

MS Teams is the best messaging app on the market, for businesses. Microsoft (2019) says it offers a workspace for real-time collaboration, meetings, file and app sharing, and chat.

Currently, MS Teams is being used as a day-to-day file-sharing application for siloed departments to share knowledge content. Its benefits of being a real-time messaging system and file streaming service allow for quick sharing of information between users. Also, MS teams is integrated with Azure Active Directory, which secures its services by SSO, single sign-on.

The business has setup teams so that each user is assigned to their departments, which allow them to access certain channels.

### 2.1.1 Limitations

MS Teams does not provide, in isolation, services required to meet the business’ desired state as described in the above gap analysis (Table 1). MS Teams has a static library structure which makes arranging and organising of knowledge difficult and near to impossible without SharePoint. This does not fit our need to have an application that enables users to manage and tag content. Additionally, MS Teams does not provide a solution for simple content sharing as you must search between each channel and chat manually meaning there is no solution for global search of files or information.

## 2.2 Dedicated SharePoint site

Microsoft SharePoint is a document management and collaboration platform that allows corporations to manage archives, documents, reports, and other content that is essential to its operational procedures.

It will be simpler for staff who are already accustomed to using SharePoint for daily document management and business activities to switch to a full knowledge management solution set up within their current environment.

### 2.2.1 Limitations

SharePoint has a search capability that, when used with wide query parameters, can be sufficient for obtaining a limited number of files, although the performance quickly falters when used with larger data sets and more specific search criteria.

Below are some limitations on searching –

* Without further customization, the only factor that can be used to filter search results is the document's age.
* The user's current site collection is the only one that may be searched.
* How administrators configured the functionality has a significant impact on the user search experience.

*(Brooks & Ryan Brooks, 2020)*

The above limitations impact the suitability of SharePoint, as a knowledge platform, due to it not achieving the desired state of global searching content. Searches are frequently for popular articles or rules, amongst others content. Although, several items have similar search phrases, there is frequently only one item that will provide the query’s answer.

## 2.3 Miro board

Miro positions itself as a collaborative digital whiteboard that is simple to use. You can arrange objects, make notes, and connect via online chat or embedded videocalls all within the software. Miro is currently used by the business as a whiteboard session workspace, allowing team members to share and collaborate on feature projects.

If we were to use Miro as a knowledge platform it will primarily aid most as a visual knowledge platform, alongside the additional uses listed below.

* For each unique activity, you can construct a separate board, or you can make a single board for the complete knowledge base.
* To implement "Navigation" across the board, use internal linking. If the navigation is done right, this will prove to be a vital element to allow for content linking.
* You can generate the content for the actual knowledge base by utilising different shapes, text, connection lines, and pen tools.

Miro board also provides access through SSO (Single-Sign on) which allows employees to access the service without using external emails outside of the business.

### 2.3.1 Limitations

An issue with using Miro, is that if it is not managed it can become unwieldy and become hard to go back to existing boards. Users have limited structure or templates to fill in content within Miro board, which does not fit in with our desired state manageable, knowledge resources. Furthermore, this solution would also make it difficult to analyse and report on current knowledge content, shared within the system.

Additionally, Miro board doesn’t not have searching capabilities, other than the name of the board, which further does not fit with the desired state of content searching, meaning users must open multiple boards before finding the resources they require.

## 2.4 Nuclino

Teams may share and edit documents in real-time with the aid of the knowledge management and collaboration platform Nuclino. It has a variety of capabilities, such as visual editors and third-party connectors, and is quite simple to use.

By creating material from scratch or using a template from their extensive library, it is easy to adapt to team requirements. This fits with our desired state of having a manageable knowledge resource structure and would allow the business to analyse knowledge resources across the business.

Furthermore, the effective search function rapidly displays pertinent results, and the phrases are highlighted in the results, making it simpler to locate desired information (Hero, 2021).

The system also supports real-time collaboration and commenting, by showing group changes in real time. This also achieves our desired state of collaborative feedback, by allowing employees to comment and share thoughts on knowledge resources shared by the author (Hero, 2021).

### 2.4.1 Limitations

A limitation of Nuclino is that it is a novel tool outside of the business infrastructure, meaning knowledge is hosted outside of the business environment, which would limit any internal confidential content being shared.

Nuclino uses a pricing structure (£4.00 per user, per month) which could be a limitation due to budgetary constraints, going against the desired need for the knowledge platform to be accessible companywide. If the business were to implement Nuclino, it would cost £40,800.00 per year on licenses (£4.00 x 850 employees).

## 2.5 Knowledge Sharing in Digital Platform Ecosystems: A Textual Analysis of SAP's Developer Community

The paper "Knowledge Sharing in Digital Platform Ecosystems: A Textual Analysis of SAP's Developer Community" by Martin Kauschinger examines the knowledge-sharing practices within the SAP developer community. The study uses a textual analysis of posts and comments from the community forum to explore the nature of knowledge sharing and the factors that influence it.

The results of the analysis indicate that the SAP developer community is an active and diverse platform for knowledge sharing. The study identifies several themes related to knowledge sharing, including problem-solving, collaboration, and knowledge exchange. The analysis also reveals that knowledge sharing is influenced by factors such as trust, reputation, and the availability of resources.

Overall, the paper suggests that digital platform ecosystems can be effective in facilitating knowledge sharing, and that factors such as trust and reputation are important for building a strong community of knowledge sharers. The findings of this study have important implications for the design and management of digital platform ecosystems that aim to promote knowledge sharing and collaboration.

The key points to take away from the paper "Knowledge Sharing in Digital Platform Ecosystems: A Textual Analysis of SAP's Developer Community" by Martin Kauschinger are:

1. The SAP developer community is an active and diverse platform for knowledge sharing.
2. Knowledge sharing in the community is facilitated by factors such as trust, reputation, and the availability of resources.
3. The study identifies several themes related to knowledge sharing, including problem-solving, collaboration, and knowledge exchange.
4. Digital platform ecosystems can be effective in facilitating knowledge sharing and collaboration.
5. The findings have important implications for the design and management of digital platform ecosystems that aim to promote knowledge sharing and collaboration.

Overall, the paper highlights the importance of creating a strong community of knowledge sharers in digital platform ecosystems and provides insights into the factors that influence effective knowledge sharing practices.

## 2.6 Knowledge Sharing in Digital Platform Ecosystems: A Textual Analysis of SAP's Developer Community

The paper titled "Social Media for Knowledge-Sharing: A Systematic Literature Review" aims to identify and analyze the existing literature on the use of social media platforms for knowledge sharing. The authors conducted a systematic literature review by selecting 39 relevant studies published between 2008 and 2017.

The paper finds that social media platforms can enhance knowledge sharing in several ways, such as providing a collaborative environment for sharing ideas, facilitating informal learning, and enabling access to a wide range of expertise. The paper also identifies several factors that can influence the success of social media for knowledge sharing, such as organizational culture, technology adoption, and the motivation of users.

The paper concludes that social media platforms can be effective tools for knowledge sharing, but their success depends on a range of factors. The authors suggest that organizations need to develop strategies for implementing social media platforms and integrating them into their knowledge-sharing practices to achieve their full potential. They also suggest the need for further research in this area to better understand the role of social media in knowledge sharing.

The key points of the paper "Social Media for Knowledge-Sharing: A Systematic Literature Review" are:

1. Social media platforms can enhance knowledge sharing by providing a collaborative environment for sharing ideas, facilitating informal learning, and enabling access to a wide range of expertise.
2. Success of social media for knowledge sharing depends on several factors such as organizational culture, technology adoption, and user motivation.
3. Organizations need to develop strategies for implementing social media platforms and integrating them into their knowledge-sharing practices to achieve their full potential.
4. Further research is needed in this area to better understand the role of social media in knowledge sharing.

## 2.7 Comparison of technologies

Following is a comparison table comparing four various systems/tools delivering that exist today in the market.

|  |  |  |
| --- | --- | --- |
| Systems/Tools | Desired Features | Gap Features/Negatives |
| Team on MS Teams | * Secure access with SSO * Quick sharing of information | * Static library structure makes it difficult to rearrange/organise knowledge * No simple global search |
| Dedicated SharePoint site | * Secure access with SSO * Currently implemented across the business | * Limitations on searching * Static library structure – using tree navigation |
| Miro board | * Easy to add ideas to and collaborate on * Multiple content types - shapes, text, connection lines, and pen tools * Secure access with SSO | * External tool – separate to MHR’s infrastructure * Very much a whiteboard online – so they can get unwieldy and hard to go back to * Can’t search multiple boards (no global search) |
| Nuclino | * Real-time collaboration * Commenting and content history * Content templating | * Completely new tool, separate to MHR’s infrastructure. * Content hosted outside of MHR datacentre. * High price for budget |

## 2.6 Summary

The above review and comparison of different systems/tools against their suitability highlights there is not a current solution to fit all the desired states identified within the gap analysis. Whilst Nuclino provides all desired states, its highly priced subscription plan and separate cloud service solution for storing data is not ideal.

Therefore, this new system will be built inhouse, allowing for controlled outcomes of the solution’s requirements and features. This will also enable cohesive services for employees to use and provide a cloud application allowing employees to manage, tag, and search for content. This will encourage reduction of silos due to increased collaboration, sharing of ideas/knowledge, and between department idea generation.

### 2.6.1 Challenges

Some challenges to consider in the solution:

* **People don’t share new knowledge**: The repository needs to be added to regularly to ensure knowledge doesn’t stay in personal notes, OneDrive documents or within teams, therefore it needs to be accessible and easy to use.
* **People can’t find knowledge**: The tool needs to have a good mechanism for searching so that discoverability of current knowledge is easy.
* **Knowledge needs to stay in-house**: The tool needs to have secure access for MHR internal employees only (e.g. use of single sign on).
* **People will want the solution to fulfil other requirements**: When creating the solution, ensure that it doesn’t try to fulfil too wide a scope – first and foremost it is an easy-to-use repository of knowledge for MHR.

# Project management

## 3.1 Scope

This project is to develop a cloud-based central knowledge that reduces duplication of effort, increases collaboration, sharing of new ideas/knowledge, and idea generation between departments. The central repository will allow users to manage, feedback and share content between employees of MHR.

## 3.2 Schedule plan

The project will begin at the start of November 2022 with an estimated end date mid-April 2023, a total project time of 4 month. The project timescales include additional time to allow for contingency of work items, if in case additional tasks are needed or existing tasks needs more time to be completed. Work items will be planned into each timebox before it starts, following the agile pattern with a two-week timeboxes. This way, the capacity of each team member for the following sprint is known, and enough work can be planned with contingency time for issues or bugs that may prevent the team from progressing the work items for the project.

## 3.3 Resource planning

The project plan tools details and outline the tasks and scope of the project with each team member assigned to it. The engineers, within the business, already possess the necessary technical expertise and skills for the project, both analysis and use of technology, this means no outside resources will be required to complete this project. It is crucial that at least one executive level participant attend the design sessions of the project.

Below describes the development project team, the team only consist of 5 members –

|  |  |
| --- | --- |
| Job titles | Skills |
| Lead Full-stack Engineer | * Lead project manager for team * Full stack (Angular & C#) * Architecture design |
| Software Engineer (.Net) | * NET (C#) * SQL Database * Microservices design * CI/CD pipelines (backend services) * Automated testing |
| Frontend Web developer | * Angular (Typescript) * HTML/CSS3 * CI/CD pipelines (frontend application) * Automated testing |
| UX Designer | * Design UX for requirements |
| Business analyst | * Details requirements * User acceptance testing * Senior Stakeholder |

## 3.4 Gantt Chart & PERT Diagram

A Gantt chart resource plan has been created and can be accessed within the file *‘COMP37005\_Project\_Management\_Plan\_GANNT.mpp’*, which can be found in annex 1.

A PERT diagram can be also found within the fille *‘COMP37005\_Project\_Management\_Plan\_PERT.mpp’* via annex 1, shows the dependencies of the project.

The Gantt chart project plan details the critical tasks with project, such as initial analysis and requirement gathering phase, foundations for identifying the specifications for scope, and then the development timeline for phase 1, which consists of 3 timeboxes. Tasks has been mapped out with their dependencies which has created the critical path for the project. Some tasks can be started and finished in parallel with others, such as timebox development work. Within timebox 1, tasks 4.1.2.1, 4.1.2.2, 4.1.2.3 can be ran in parallel of each as its solo dependency Is the refinement of the work (4.1.1.1) and are not dependent on each other and ran in isolation with different team members. Another example of parallel tasks is 2.6.1 and 2.6.1, the two modelling tasks, these can be run in parallel because are not dependent on each other as well and requirements of the project being defined before beginning.

## 3.5 Analysis of Legal, Social, Ethical and Professional Issues

### 3.4.1 Legal Issues

#### 3.4.1.1 Licensing

Currently the project is only for internal use and is not required to be licenced. If the business wants to transition the project to customer consumption, the product will require licencing. The rights of the creator and user of a piece of software are outlined in the software licence. It outlines the software's usage restrictions and how to pay for it. *(Lutkevich & Lebeaux, 2021)*.

If project would become open source and require no official software licencing, it can provide a MIT licence.

#### 3.4.1.2 Intellectual property of content

Intellectual property infringement damages can range from an injunction to a judgement for several million pounds. The knowledge platform be legally taken down if it violates a patent, trademark, copyright, trade secret, or trade dress.

A simple prevention for this is all employees must reference any knowledge shared on the repository if it’s it came from an external source outside of the business.

#### 3.4.1.3 Data ownership

The product provides the feature of allowing a user to enter in information and content, because of this it falls under the GRPDR act. Data collection and storage must adhere to the regulation, and software should only request and handle the data essential for the software to operate.

Any violations must be disclosed to individuals impacted, failure to comply with this can result in significant fines and a negative company reputation. *(Art. 5 GDPR – principles relating to processing of personal data 2021)*

#### 3.4.1.4 Source control

All the code of the project will be stored away within source control, Azure DevOps. It’s very important to ensure we use a verified source control provider as we want to code to be secure. The implementation is not open source, meaning no external individual will be given read or write access to the repository. This will protect the IP (Intellectual Property) of the product.

Only verified personnel within the business, using Azure Active Directory, will have permission on the repository.

### 3.4.2 Social Issues

#### 3.4.2.1 Free text content

A feature of the product is allowing the user to enter free text as content to share. This could lead to users posting insensitive texts and content around the system that would offend other users which will lead to social issues within the business. Social issues are guarded by company polices on communication and interaction of software between employees.

Content admins are allowed to manage all content around the system, this will allow the admin to review all free text shared on the system and remove any inappropriate content.

#### 3.4.2.2 Content sharing

Another feature of the product is allowing users to share external content within the system. This could lead to users sharing inappropriate work content across the system, which again could lead to users being offended and social issues.

A prevention for this is content admins will have a list of verified domains, and users can only share information from a domain which has been verified.

### 3.4.3 Ethical Issues

#### 3.4.3.1 Commenting for relative feed (gamification)

It is possible that users will try and force other users to comment on their knowledge content so that their post is more likely to come up in other users’ feeds. This is very unethical as users will try and exploit the commenting system for gamification. For this reason, commenting has no effect on order of post on other users’ feeds.

#### 3.4.3.2 BCS (Code of conduct)

Software developers are required to abide by a code of ethics and a set of guidelines when creating, designing, and maintaining software.

Below are some examples from BCS Code of conduct for making software for everyone *(BCS Code of conduct)* –

* have respect for the environment, other people's well-being, their privacy, security, and public health.
* respect the legal rights of others as they deserve.
* conduct your professional duties without prejudice based on a person's gender, sexual orientation, marital status, race, colour, nationality, ethnicity, religion, age, or disability, or any other condition or requirement.
* encourage equal access to the advantages of IT and work to advance the participation of all societal segments whenever chances present themselves.

#### 3.4.3.3 Plagiarism

With the sharing content feature, a user might try and plagiarise knowledge content as their own. Copying someone else’s work is unethical because it does not give credit to the author which can lead to false impression about people skills and callabilities.

Content admins can verify knowledge content around system to ensure it’s not been plagiarised. An extra feature, if in scope, is to introduce a Turnitin like software which will check the content automatically as a part of the post feature.

### 3.4.4 Professional Issues

#### 3.4.4.1 Business related content

The content for this system should be limited to business or business skills related information to aid employees for upskilling and training purposes. Other content is not desired as this is a business tool for internal use.

## 3.5 Risk Assessment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Description | Likelihood | | Impact | Severity | Owner | Mitigation |
| Scope of project might change | | Medium | Medium | Medium | Business Analyst | Ensure all scope is defined in detail via the design workshops and all stakeholders are involved. |
| Lack of communication leads to unclear thinking and misunderstanding. | | Low | Medium | Medium | Team | Create a communication plan that details the timing, purpose, and target audience for each communication. Timebox reviews will allow the team time to feedback on the timebox. |
| Deficient testing leads to large post go live bug list. | | High | High | High | Lead Engineer | Make sure the tester creates test cases and quality checks, and that they guard the window for testing and quality assurance. |
| Small team absences/sickness might delay project. | | Medium | High | High | Lead Engineer | Additional time buffer added to guard any unplanned absences or sickness. |
| Undefined project needs and purposes | | Medium | High | High | Business Analyst | Finish the business case and make sure the project charter's purpose is clearly stated. |
| Unexpected tasks that need to be accommodated | | Low | Medium | Medium | Lead Engineer | Verify each plan and each quantity survey. Record all planning assumptions before the project begins. |

# Requirement analysis

## 4.1 Research and Discovery

Requirements will be acquired through a joint application design (JAD) session to match the business needs found in the GAP analysis. As this is a fresh project, with no existing system to replace, the JAD session will concentrate on developing new needs without comparison to an existing system. The JAD session participants can communicate and provide a better grasp of the aims and goals, relative to their abilities and knowledge. This should clear up uncertainties regarding the deliverable’s ultimate goals, objectives, and expectations.

In addition to a JAD session, surveys will be created and provide to selected individuals within the business, in each siloed department. Each survey will gather information of key stakeholders regarding how knowledge is shared within their departments and team, with the aim of analysing variations and similarities. This information will be used in brainstorming and workshop sessions to better understand system users, and to produce applicable solutions which allow all types of knowledge to be shared companywide.

Some questions to ask people to understand the current situation from their points of view and add more to this section:

* Where do you currently store notes, papers, links etc
* When you attend an event, do you share your notes? With who?
* Do you share any knowledge (links/papers/documents) with other departments?
* What would be useful to you if you could search in another department’s notes/documents?
* When people share articles/videos with you, do you have the time to read/watch them?

If these requirement elicitation techniques are achieved, it will reduce the ‘#1 - Requirements are not fit’ risk, within the risk analysis.

The JAD session will be divided of three days. Each survey will be given out on a conduct needs analysis, to create appropriate questions.

* Day 1 – Survey feedback review and use cases
* Day 2 – Requirements analysis, brainstorming and workshop
* Day 3 – Specification design session

A senior stakeholder will be present within the JAD sessions, who will be a project representative and have final say on related decisions. Said stakeholder could be the CEO, CIO, or project manager (Davidson, E.J., 1999). Attendees will be made to feel as if they are a part of the process and the project objectives stated. Should initial worries be voiced, the stakeholder should strive to alleviate them.

## 4.2 Software Development Methodologies

There are two approaches that might be employed for this project: waterfall and agile.

### 4.2.1 Waterfall

A traditional model used in the system development life cycle to design a system with a linear and sequential approach is the waterfall model. Because the model progresses methodically from one phase to the next in a downward direction, it is known as a waterfall model. The output of one phase is utilised as the input for the following phase in this model, which is broken up into several phases. There is no overlap between phases; each must be finished before the next one begins.

Since each step is preplanned in a linear order, one drawback of the waterfall project management method is that it is not very adaptable. Any alteration in stakeholder objectives or demands will throw off the arrangement and necessitate a revision or maybe a completely new blueprint. For knowledge-based tasks, like computer programming, waterfall project management performs less well. However, waterfall workflows can be simply duplicated for upcoming, comparable activities, making up for its lack of flexibility. *(What is Waterfall Project Management? 2016).*

### 4.2.2 Agile/Scrum

Agile methodology employs the "Waterfall" style of software development and focuses on communication and results. The concept focuses on self-management, short timescales, and flexible planning, enabling system improvements over time.

It eliminates the possibility of timelines that span months or years and lessens the likelihood that projects would fail because of errors made earlier on. Each phase is kept brief—under 4 weeks—which enables the system to advance quickly. Even though there is less documentation, the project places a strong emphasis on putting the key stakeholders needs first by allowing teams and employees to collaborate closely with them.

Scrum is the component of Agile approach that we will focus on. Most firms utilise this model since it offers a simple framework for processes. In Scrum, a product owner creates a prioritised wish list known as the product "backlog." From this backlog, teams, which are often small, will take bits of the backlog and produce a sprint. A sprint is a period (between 2-4 weeks) during which this item on the backlog will be evaluated, created, and integrated into the larger system. The teams will gather each day for a meeting called a "daily scrum," which tracks progress on the sprint and is run by a Scrum Master to keep the team on task. When the sprint is finished, they review and deploy the changes, then a new sprint is started.

An agile strategy might be used to increase the project's adaptability to changes and start delivering work sooner so that stakeholders could inspect it and provide input, ultimately giving end consumers a richer result. *(Denning, 2022)*

### 4.2.3 Chosen Development Method

The Agile methodology makes use of short, fixed-length iterations that concentrate on a small number of components of change. Each iteration is produced, tested, and distributed using prototypes, which allows for quick user feedback to inform current and subsequent revisions, this will allow to understand and create the correct solution for internal use.

This internal system is very heavily prototyped, allowing the requirements to be abstracted and divided into smaller timeboxes with a minimum number of features would benefit flexibility of change and adapting of features. *(Alberto Sillitti, 2011)*

One of the main principles of Scrum is that the business gives the scrum team authority to complete the task. The scrum team develops into a self-sufficient division of the company. This will allow the team more control on the work, not rushing each task to a deadline.

Although Agile approaches first appear to result in greater development costs and extended project timeframes due to the crucial role conversations across teams/companies play, this gradual process empowers users and promotes collaboration.

Requesting software specifications and requirements from key stakeholders using the waterfall process meant they had to specify exactly what they needed up front. Their specifications were outlined in a document that the development team and testers used to make sure they delivered exactly what the client had requested, without adding or removing anything.

This strategy assumed that customers knew what they wanted from the start. However, this was infrequently the case, and customers frequently requested changes to the specification after changing their minds, this increases our risk of “requirements to not fit.” The impact on the delivery date and who would be responsible for the cost of the change were then discussed.

The project will be using the Agile Scrum structure to plan, research, and create this system. Agile is a great lightweight framework for small, straightforward projects with compressed timelines.

## 4.2 Functional and Non-Functional requirements

Below I will analyse the requirements given by the identified stake holders, from there I will split these requirements into Functional and Non-Functional requirements. Functional consist of features that the system or data must contain. Non-Functional is aspects of the system which is general behaviours that the system needs.

I will detail each requirement and reference from the requirement.

|  |
| --- |
| Functional Requirement |
| Process-oriented |
| The system must allow a user can add content into the system via content templates – links, word documents, free-text |
| The system should allow a user to add a summary for content for such knowledge resources as articles links |
| The system must user be allowed to tag their content with different subjects, free text |
| The system must display a content feed where users can access the latest and relevant content |
| The system should have a search box where users can enter titles and tags, displaying the results. |
| The user should be allowed to add multiple comments and replies on content. |
| A user must be able to log-in using their work Microsoft account |
| The system could provide profanity filtering on content posts. |
| Content can be scraped to find key terms for easier searching. |
| Information-Oriented |
| Different types of content – external links, links to documents in SharePoint/OneDrive, pdfs, videos, text |
| Summaries to content (so that people don’t need to read/watch the whole piece) |
| Non-Functional Requirement |
| Operational |
| Cloud application accessed via a browser |
| 99% Availability as an SLA |
| Failover to second region if primary region fails |
| Performance |
| Files should be able to be uploaded quickly |
| When there are more than 10,000 concurrent users, the website should load in three seconds. |
| User login should take no longer than 10 seconds |
| API response times should no greater than 2 seconds |
| Security |
| Uses OAuth with Azure AD |
| All Http requests are SSL secured |
| No sensitive data which may be used to identify a user or individual, as per the General Data Protection Regulations (GDPR) |

Table 4 - Functional and non-functional requirements

## 4.3 Use Case Analysis

The system's primary functionality is represented by the three use cases listed below.

### 4.3.1 Use Case #1

|  |  |
| --- | --- |
| Use Case Name | A user adds knowledge content |
| ID | 1 |
| Priority | High |
| Actor | Subject matter expert (user) |
| Description | The user selects content template to fill (links, papers, books, free-texts). Once form is completed, the user will click save and publish, this will then persist this knowledge resource in a database. |
| Trigger | The user wants to use this system to share a piece of knowledge information with the rest of the business. |
| Preconditions | The application has verified and approved the user credentials. |
| Normal Course | User clicks on add content button  User selects content template from dropdown  User fills out content form  User submits and publishes the content |
| Postconditions | Content is persisted within database and shown on other user’s content feed.  User logs out of application. |
| Exceptions | User credentials are not valid or timed out.  Profanity filter picks up words. |

Table 5 - Use case 1

### 4.3.2 Use Case #2

|  |  |
| --- | --- |
| Use Case Name | A searches knowledge content |
| ID | 2 |
| Priority | High |
| Actor | Content consumer (user) |
| Description | The user should be able to search for the content with the system via headers and tags. |
| Trigger | The user wants to use this system to learn different knowledge around the business. |
| Preconditions | The application has verified and approved the user credentials.  Content has already been added in the system with tags. |
| Normal Course | User click on navigation icon on menu bar  Text field appears for user to write into  User press enter with text field filled  List of content results are returned |
| Postconditions | User can click content.  The user logs out of the system. |
| Exceptions | The user's login credentials are not valid.  No content in the system was added. |

Table 6 – Use case 2

### 4.3.3 Use Case #3

|  |  |
| --- | --- |
| Use Case Name | Add comments to content |
| ID | 3 |
| Priority | Medium |
| Actor | Users |
| Description | The user should be able to comment and reply on content. |
| Trigger | The user wants to use this system to provide, or ben given feedback on knowledge shared around the business. |
| Preconditions | The application has verified and approved the user credentials.  Content has already been added in the system with tags. |
| Normal Course | User navigates to content  User enters comment text field and press enters  User enters reply comment text fields and press enters (for reply to threads) |
| Postconditions | The user logs out of the system. |
| Exceptions | The user's login credentials are not valid.  No content in the system was added. |

Table 7 - Use case 3

## 4.4 MoSCoW

Below are the identified functional and non-functionals requirements MoSCoW.

The term MoSCoW stands for must-have, should-have, could-have, and won't-have (or will not have right now) for the project.

|  |  |  |  |
| --- | --- | --- | --- |
| MUST | SHOULD | COULD | WON’T |
| Content templates | Content feed | Profanity filtering | Content scraping |
| Summaries | Commenting |  |  |
| SSO – AD | Tagging |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 4.5 User stories

|  |  |  |  |
| --- | --- | --- | --- |
| Title | User Story | Story points | Acceptance Criteria |
| Build out basic python console applications for nodes | As a user I want to open the python console applications for Warden, Message brokers, Producers and Consumers | 3 | * Console for warden with Open TCP Socket * Console for broker with Open TCP Socket Console for consumer with Open TCP Socket * Console for producer |
| Create a topic queue data structure to store data (partition) | Create a topic within Message broker and queue data structure to store data (partition) | 2 | * Can create topic instance within broker * Can create partition instance within topic |
| SQLite database structure | Create SQLite database structure within warden using ERDs | 2 | * SQLite schema matches ERD |
| Broker to register its network information | As a user I want to start a broker and have It register within the cluster and stored within the warden. | 2 | * Broker starts up * Send IP and port to warden * Warden saves data in DB |
| Producers and consumers to send and receive data | As a user I want to start a publisher and push message with a topic into the broker.  As a user I want to start a consumer and pull data from the broker with a topic. | 5 | * Publisher starts up * Select topic, enter message and press send * Consumer starts up * Select topic and start receiving messages |
| Users create topics | As an admin user I want to be able to create topics within the warden for the cluster. | 3 | * Within Warden menu, “Create topic” * Enter topic name and number of partitions * Multiple partitions across different brokers. |
| Consumer group offsets | As a user I want to have a grouping for my consumers, so I can have multiple consumers pulling from the same topic. | 3 | * Enter consumer group when consumer app starts up * Don’t pull the same data from same topic when in the same consumer group |
| Create test application | As a user I want to be able to have a user interface application I can create messages and push them to a cluster and see the stats of how fast they were received by consumer | 3 | * Test application with basic GUI * Select topic, enter message and number of times to send message * See stats on how fast the messages were received |
| Security protocol for encryption and key access | As a user I want my system to be secured and not open to external threats outside of the cluster. | 3 | * Create access key for nodes to use * Create encryption for messages on inbound and outbound of broker. |

Below are the MoSCoW requirements converted into User stories. A user story is a casual, generic explanation of a software feature written from the end user's perspective. These stories will be pointed with the Fibonacci sequence.

# Design

# Implementation

## 6.1 Coding standards

Coding standards assist in the construction of less complex software programmes, which reduces. errors. If software engineering programming standards are followed, the code is consistent and easy. to maintain. This is since anyone may comprehend it and change it at any time.

### 6.1 SOLID

Single responsibility principle, open-closed principle, Liskov substitution principle, interface segregation principle, and dependency inversion principle are all acronyms for five essential design principles.

#### 6.1.1 Single Responsibility Principle

The SRP is based on the idea that each class, module, or function in a programme should have a single responsibility or purpose.

#### 6.1.2 Open Closed Principle

Classes should be open to extension but closed to modification, according to the Open-Closed Principle.

#### 6.1.3 Liskov Substitution Principle

Subclasses should be interchangeable with their base classes, according to the Liskov Substitution

Principle.

#### 6.1.4 Interface Segregation Principle

The Interface Segregation Principle is about keeping the interfaces separate, and segregation involves keeping things apart.

#### 6.1.5 Dependency Inversion Principle

The Dependency Inversion Principle emphasises that instead of concrete classes and functions, our classes should rely on interfaces or abstract classes.

### 6.2 Design patterns

In software engineering design patterns are common solutions that solves occurring problems in software design. A design pattern is not a finished design that can be used within every implementation of a piece of software but it’s a guide/template that can help you solve your problem (GeeksforGeeks. 2021.).

#### 6.2.1 Factory Pattern

The Factory Method design pattern is a creational design pattern that provides an interface for producing objects in a superclass while allowing subclasses to choose the type of objects created.

#### 6.2.2 Adapter Pattern

The adapter pattern acts as a link between two interfaces that are incompatible. This design pattern is classified as a structural pattern since it integrates the capabilities of two separate interfaces.

A single class is responsible for joining the capabilities of separate or incompatible interfaces in this design.

#### 6.2.3 Provider Pattern

Provider pattern allows for implementation of components to be introduced easily within a system. The provider object will have abstract logic or implementation that can be genetically referenced by other objects.

# Results

# Conclusion

# References

*What is Microsoft teams? (2019). Available at: https://support.microsoft.com/en-us/topic/what-is-microsoft-teams-3de4d369-0167-8def-b93b-0eb5286d7a29 (Accessed: October 22, 2022).*

*What is Waterfall Project Management?* (no date) *Wrike*. Wrike. Available at: https://www.wrike.com/project-management-guide/faq/what-is-waterfall-project-management/ (Accessed: October 25, 2022).

Lutkevich, B. and Lebeaux, R. (2021) *What is a software license? everything you need to know*, *SearchCIO*. TechTarget. Available at: https://www.techtarget.com/searchcio/definition/software-license (Accessed: November 1, 2022).

*Art. 5 GDPR – principles relating to processing of personal data* (2021) *General Data Protection Regulation (GDPR)*. Available at: https://gdpr-info.eu/art-5-gdpr/ (Accessed: November 1, 2022).

*BCS Code of conduct* (no date) *BCS*. Available at: https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/ (Accessed: November 1, 2022).

Hero, M. (2021) *Nuclino - the underdog knowledge base tool you should try*, *Markup Hero - Blog*. Markup Hero - Blog. Available at: https://markuphero.com/blog/nuclino-knowledge-base-tool-review/ (Accessed: November 1, 2022).

Brooks, R. and Ryan Brooks Product Evangelist at Netwrix Corporation (2020) *Using SharePoint as a Knowledge Management System: PROS and cons*, *Using SharePoint for Knowledge Management: Pros and Cons*. Available at: https://blog.netwrix.com/2020/02/18/using-sharepoint-for-knowledge-management-pros-and-cons/ (Accessed: November 2, 2022).

Aviva (2019) “Great knowledge base for capturing information,” *Capterra*. Available at: https://www.capterra.co.uk/reviews/174926/nuclino (Accessed: November 1, 2022).

Davidson, E.J., 1999. Joint application design (JAD) in practice. The Journal of Systems and Software, 45 (3), 215-223. (Accessed: October 23, 2022).

Alberto Sillitti, 2011. Agile Processes in Software Engineering and Extreme Programming. 12th International Conference, XP 2011 (Accessed: October 29, 2022).

# Annex

## 6.1 Annex 1

GANNT Chart

Attached file named “COMP37005\_Project\_Management\_Plan\_GANNT.mpp”

PERT Diagram

Attached file named “COMP37005\_Project\_Management\_Plan\_PERT.mpp”