Requirements Gathering through AI-based tools for better Knowledge Management/Portfolio Management.

MCS

Author

Arisha Masood Khan

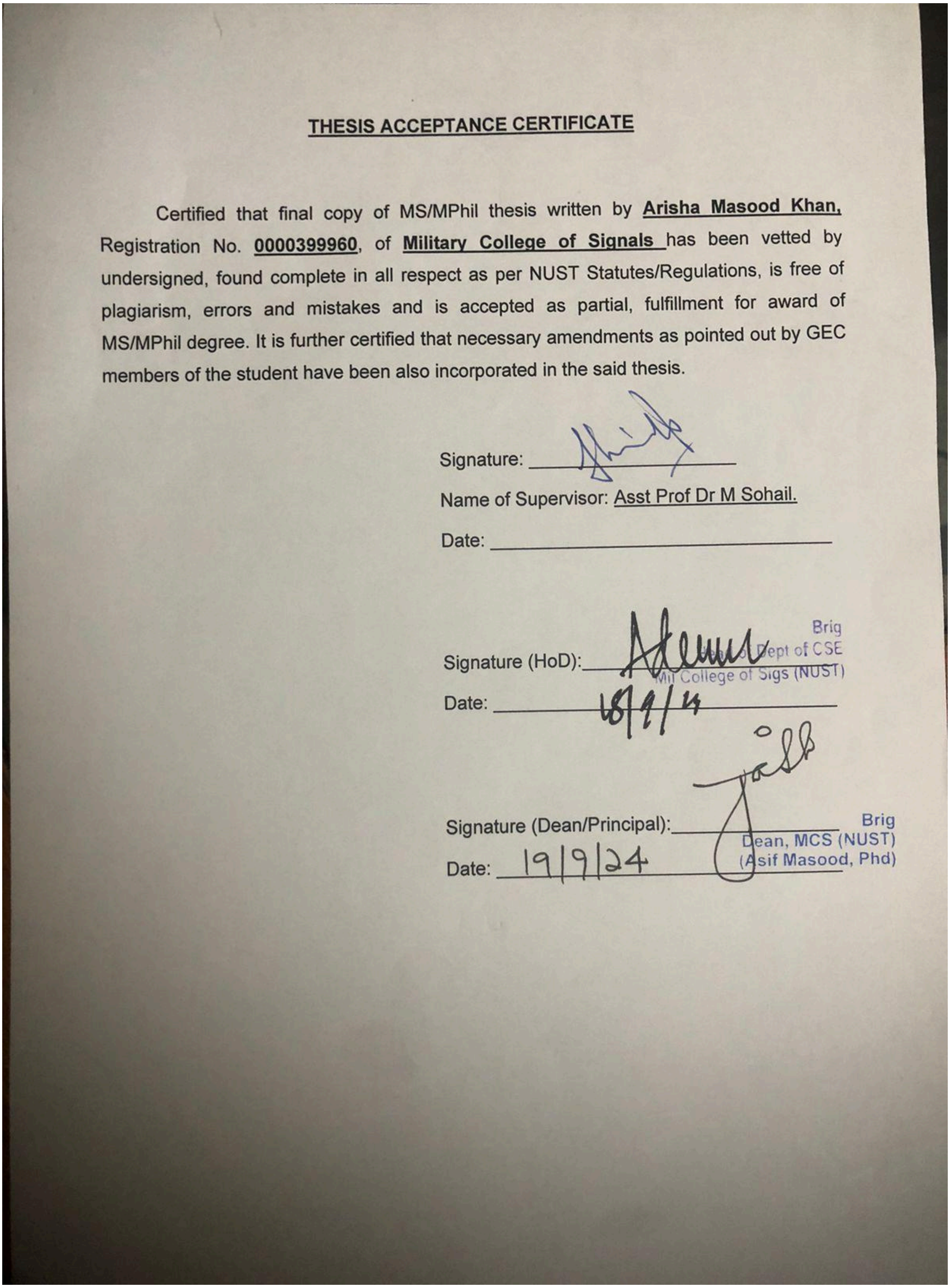
Registration No: 00000399960

Supervisor:

Asst. Prof. Dr. Muhammad Sohail

A thesis submitted to the faculty of Computer Software Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan in partial fulfillment of the requirements for the degree of Masters in Software Engineering

(September 2024)



THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by **Arisha Masood Khan**, Registration No. **0000399960**, of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: [illegible]

Name of Supervisor: Asst Prof Dr M Sohail.

Date: ______

Signature (HoD): [illegible]

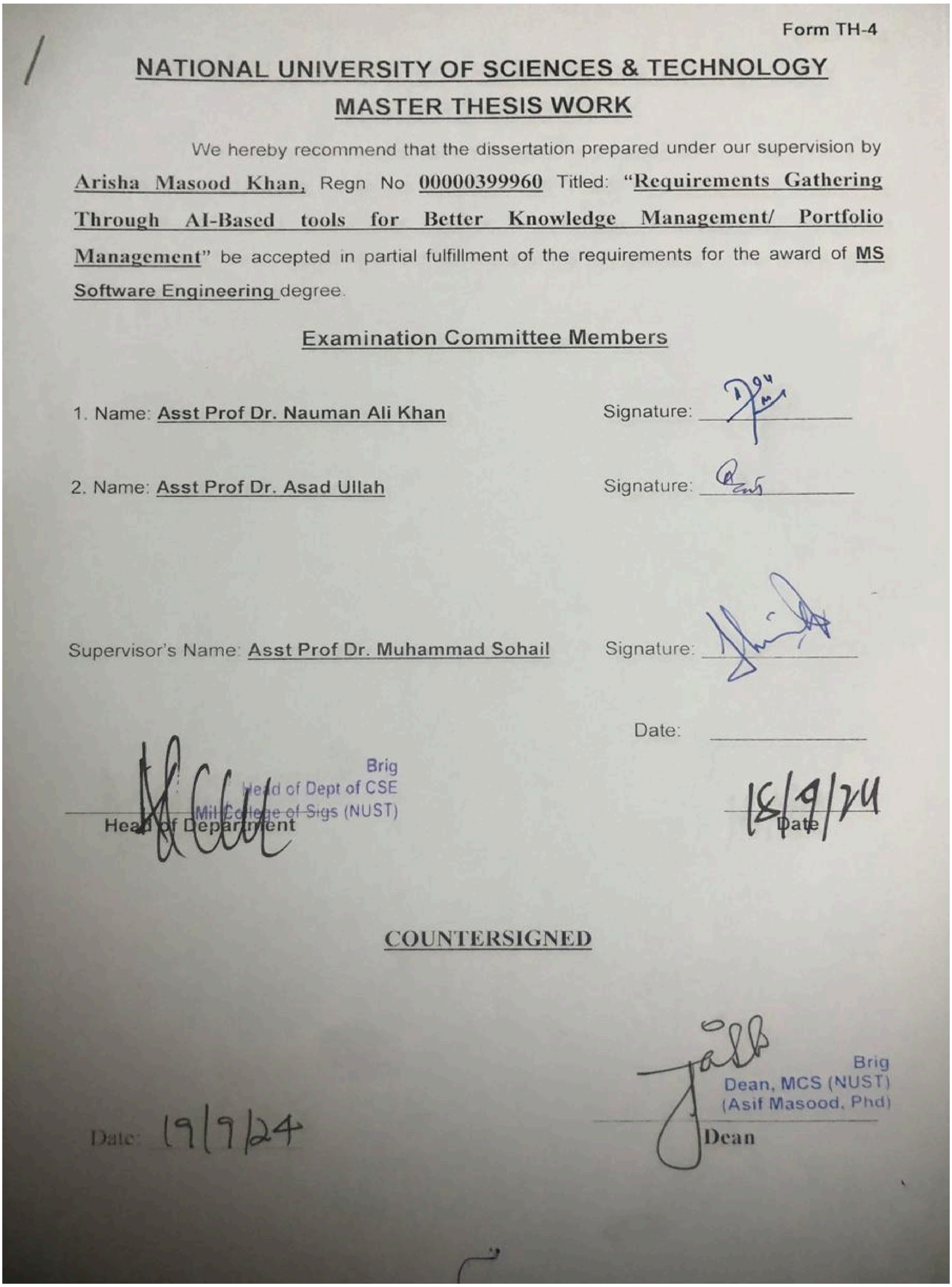
Brig
Head of Dept of CSE
Mil College of Sigs (NUST)

Date: 18/9/24

Signature (Dean/Principal): [illegible]

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

Date: 19/9/24



/

Form TH-4

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by **Arisha Masood Khan,** Regn No **00000399960** Titled: "**Requirements Gathering Through AI-Based tools for Better Knowledge Management/ Portfolio Management**" be accepted in partial fulfillment of the requirements for the award of **MS Software Engineering** degree.

Examination Committee Members

1. Name: **Asst Prof Dr. Nauman Ali Khan** Signature: [illegible]

2. Name: **Asst Prof Dr. Asad Ullah** Signature: [illegible]

Supervisor's Name: **Asst Prof Dr. Muhammad Sohail** Signature: [illegible]

Date:

[illegible]

Brig
Head of Dept of CSE
Mil College of Sigs (NUST)

Head of Department

18/9/24

Date

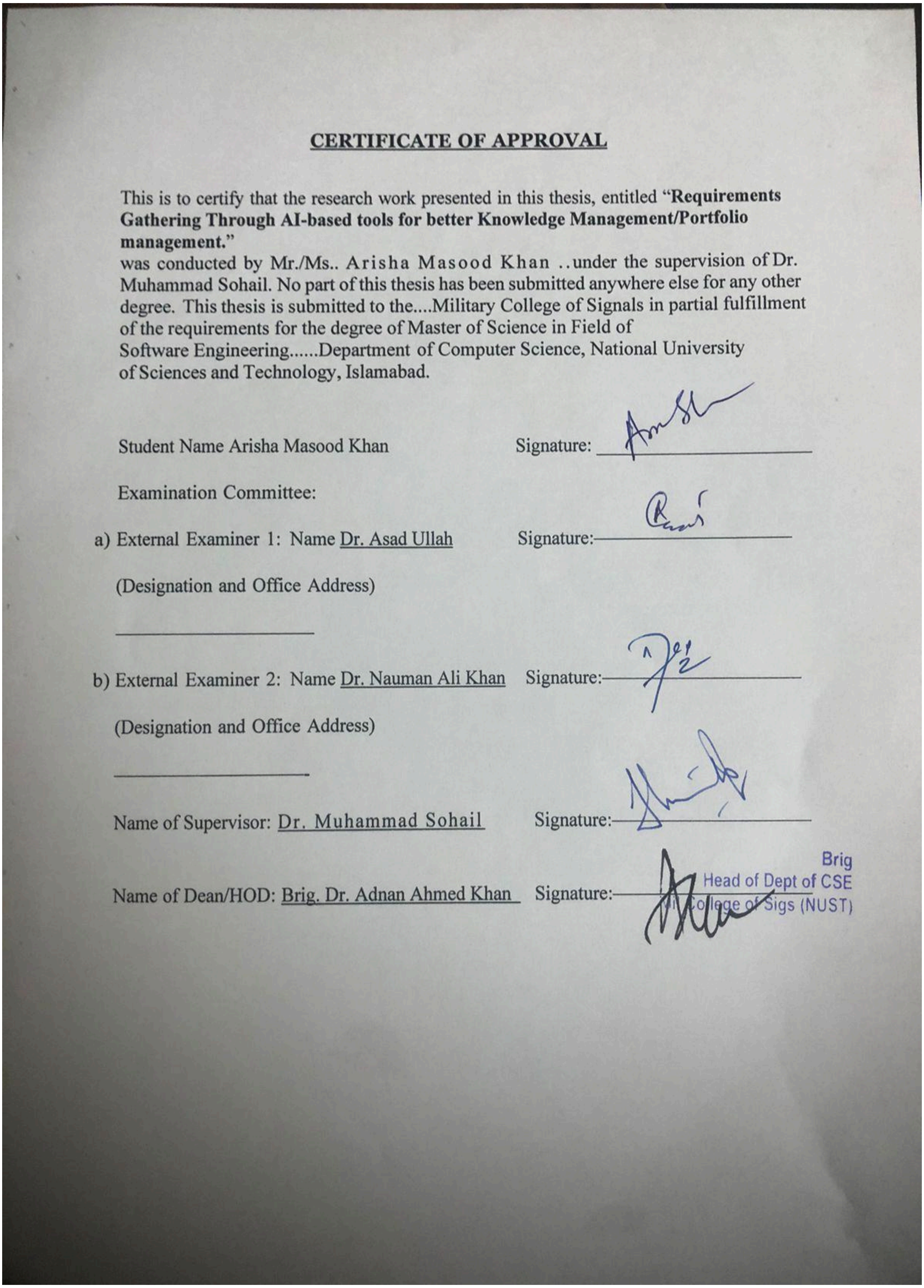
COUNTERSIGNED

Date: 19/9/24

[illegible]

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

Dean



CERTIFICATE OF APPROVAL

This is to certify that the research work presented in this thesis, entitled **"Requirements Gathering Through AI-based tools for better Knowledge Management/Portfolio management."**
was conducted by Mr./Ms.. Arisha Masood Khan ..under the supervision of Dr. Muhammad Sohail. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the....Military College of Signals in partial fulfillment of the requirements for the degree of Master of Science in Field of Software Engineering......Department of Computer Science, National University of Sciences and Technology, Islamabad.

Student Name Arisha Masood Khan Signature: [illegible]

Examination Committee:

a) External Examiner 1: Name Dr. Asad Ullah Signature: [illegible]

(Designation and Office Address)

____

b) External Examiner 2: Name Dr. Nauman Ali Khan Signature: [illegible]

(Designation and Office Address)

____

Name of Supervisor: Dr. Muhammad Sohail Signature: [illegible]

Name of Dean/HOD: Brig. Dr. Adnan Ahmed Khan Signature: [illegible]

Brig
Head of Dept of CSE
Mil College of Sigs (NUST)

AUTHOR'S DECLARATION

I **Arisha Masood Khan** hereby state that my MS thesis titled "**Requirements gathering through AI-based tools for better knowledge management/Portfolio management**" is my own work and has not been

submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Student Signature:——— [illegible] ———

Name: Arisha Masood Khan

Date: 23rd September 2024.

PLAGIARISM UNDERTAKING

I solemnly declare that the research work presented in the thesis titled "**Requirements gathering through AI-based tools for better knowledge management/Portfolio management.**" is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature:—— [illegible] ——

Name: Arisha Masood Khan

Date: 23rd September 2024

DEDICATION

Dedicated to my grandmothers: Mrs Razia Masood and Mrs Salima Ahsan, whose unconditional love enabled me to reach this milestone, my exceptional parents:

Mr. Mazhar Masood & Mrs. Asma Mazhar

whose tremendous support, encouragement and cooperation led me to this accomplishment, and adored siblings.

ACKNOWLEDGEMENTS

All praise and glory to Almighty Allah (the most glorified, the highest) who gave me the courage, patience, knowledge and ability to carry out this work and to persevere and complete it satisfactorily.
Undoubtedly, HE eased my way and without HIS blessings I can achieve nothing.

I would like to express my sincere gratitude to my advisor Dr. Muhammad Sohail for boosting my morale and for his continual assistance, motivation, dedication and invaluable guidance in my quest for knowledge. I am blessed to have such a co-operative advisor and kind mentor for my research.

Along with my advisor, I would like to acknowledge my entire thesis committee members for their cooperation and prudent suggestions.

My acknowledgement would be incomplete without thanking the biggest source of my strength, my family. I am profusely thankful to my beloved husband who continued to support me throughout every phase of my journey during this time and my loving children who were with me through thick and thin.

Abstract

Requirements management involves the systematic organization and monitoring of the needs and specifications of a software system throughout its development stages. This intricate process necessitates efficient teamwork, communication, and monitoring. Software requirements gathering is a critical phase in software development, influencing the success and quality of the final product. Traditional methods of requirements gathering often rely on manual processes, leading to challenges such as ambiguity, inconsistency, and incomplete specifications.
Requirements gathering is a crucial phase in software development, where stakeholders' needs and expectations are collected and documented. The process is often time-consuming, prone to errors, and requires significant manual effort.

AI-based tools have emerged to support this process, leveraging natural language processing (NLP), machine learning (ML), and other techniques to improve efficiency, accuracy, and effectiveness.

Artificial intelligence (AI) offers the potential to streamline certain facets of requirements management like traceability, Quantification and impact analysis. By leveraging machine learning algorithms, it's feasible to discern the connections between requirements and other elements like test cases, project knowledge and code.
AI can play a significant role throughout software development life-cycle (SDLC) but in this research I would like to particularly focus on how AI based requirements gathering can help in better Knowledge management and Portfolio management by enhancing requirements traceability,standardization, quantification, decision-making, and overall project prediction performance, which in turn can help an organization achieve Capability maturity model (CMMI) level 3 and 4.

Keywords: Chatbot, Semantic annotation, NLP, Requirements engineering, CMMI, Project management, Knowledge management, portfolio management, Lang-chain, Open-AI.

Table of Contents

Chapter 1
Introduction

With the advent of AI, every aspect of SDLC has been revolutionized. From the conception to the final product AI is capable of contributing significantly at each step.

One of the most important and often neglected parts of 'Project management' is the "closure" of the project in an appropriate way. Traditionally software houses and development teams consider handing off project to the client and successfully passing User acceptance test (UAT) as the final step and closure of project, whereas, in true essence ,systematically gathering the knowledge gained throughout the Software Development Life-Cycle (SDLC) and storing it in the company or team's portfolio for future use is what constitutes a successful "closure" of a project.

Traditional methods of analyzing Software Requirements Specification (SRS) document analysis rely on manual processes, leading to challenges such as ambiguity, inconsistency, and incomplete specifications.In the rapidly evolving field of software engineering, the effective analysis of Software Requirements Specifications (SRS) is crucial for ensuring successful project outcomes. Our research introduces a tool leveraging LangChain and GPT-4 to enhance SRS document analysis. This advanced tool integrates state-of-the-art natural language processing techniques to automate the extraction, classification, and evaluation of requirements from complex SRS documents. By harnessing the power of GPT-4's contextual understanding and LangChain's modular framework, our tool not only improves the accuracy and efficiency of requirements analysis but also knowledge management and portfolio management practices.

Knowledge management and project portfolio management are two important components of Closure that can help organizations achieve their strategic goals and optimize their performance.
Here is a brief explanation of each concept and why they are important.

1.1 Requirement Gathering

Requirements management is a vital aspect of software development, encompassing the systematic organization and monitoring of the needs and specifications of a software system throughout its development stages. This complex process demands seamless teamwork, effective communication, and meticulous monitoring to ensure that the final product meets the desired outcomes.

At the heart of requirements management lies software requirements gathering, a crucial phase that significantly impacts the success and quality of the final product. During this phase, stakeholders' needs and expectations are collected and documented, laying the foundation for the development process.

However, traditional methods of requirements gathering often rely on manual processes, leading to a multitude of challenges. Ambiguity, inconsistency, and incomplete specifications are common pitfalls that can arise from these manual approaches. Furthermore, requirements gathering is a time-consuming and error-prone process that requires substantial manual effort.

To overcome these challenges, it is essential to adopt efficient requirements management practices that facilitate collaboration, clarity, and accuracy. By doing so, software development teams can ensure that the needs and expectations of stakeholders are accurately captured and translated into a knowledge base.

1.2 Knowledge management and Portfolio management

1.2.1 Knowledge management

is the conscious process of defining, structuring, retaining, and sharing the knowledge and experience of employees within an organization. It helps organizations leverage their collective expertise, improve decision-making, reduce risks, and foster a culture of learning and innovation.

1.2.2 Project portfolio management

is the coordinated management of one or more portfolios of projects, programs, and other work to achieve specific strategic objectives. It helps organizations align their project investments with their strategic vision, prioritize and balance competing demands, and optimize the use of resources and capabilities based on previous data.
Therefore the Project portfolio and knowledge management can play a key role in an

organization's success by optimizing a company’s process.
Quantifying knowledge gained from a project is important for several reasons:

- It helps to measure the impact of knowledge management on project performance and outcomes, such as quality, efficiency, customer satisfaction, and innovation.
- It enables the identification and recognition of the knowledge assets and capabilities of the project team, as well as the knowledge gaps and needs that require further development.
- It also facilitates the transfer and reuse of knowledge across projects, programs, and portfolios, enhancing organizational learning and knowledge sharing culture.
- It supports the alignment of project knowledge with patterns, ensuring that the project delivers value and benefits to the organization and its stakeholders.

By applying knowledge management techniques to project portfolio management, organizations can enhance their communication and integration, learn from past successes and failures, and continuously improve their project performance based on qualitative and quantitative metrics.

1.3 Capability maturity model:

CMMI (Capability Maturity Model Integration) is a framework used to measure the maturity of an organization's processes, it comprises of 5 steps:
Level 1: Initial Characterized by ad-hoc processes, lack of standardization, and reactive decision-making. This level is akin to the "craft" stage of process development, where activities are guided by individual expertise rather than formalized procedures.
Level 2: Managed Defined by the establishment of project-level processes, with a focus on planning, tracking, and oversight. This level corresponds to the "defined" stage of process development, where processes are documented and followed, but may not be uniformly applied across the organization.
Level 3: Defined Marked by the institutionalization of standardized processes across the organization, with a focus on integration and coordination. This level is comparable to the "institutionalized" stage of process development, where processes are deeply ingrained and widely adopted.
Level 4: Quantitatively Managed Characterized by the use of quantitative data to manage processes, with a focus on performance measurement, analysis, and optimization. This level aligns with the "quantitative" stage of process development, where data-driven decision-making is the norm.
Level 5: Optimizing Defined by a culture of continuous improvement, with a focus on innovation, experimentation, and learning. This level corresponds to the "transformational" stage of process development, where processes are continually refined and adapted to achieve excellence.

CMMI Levels:

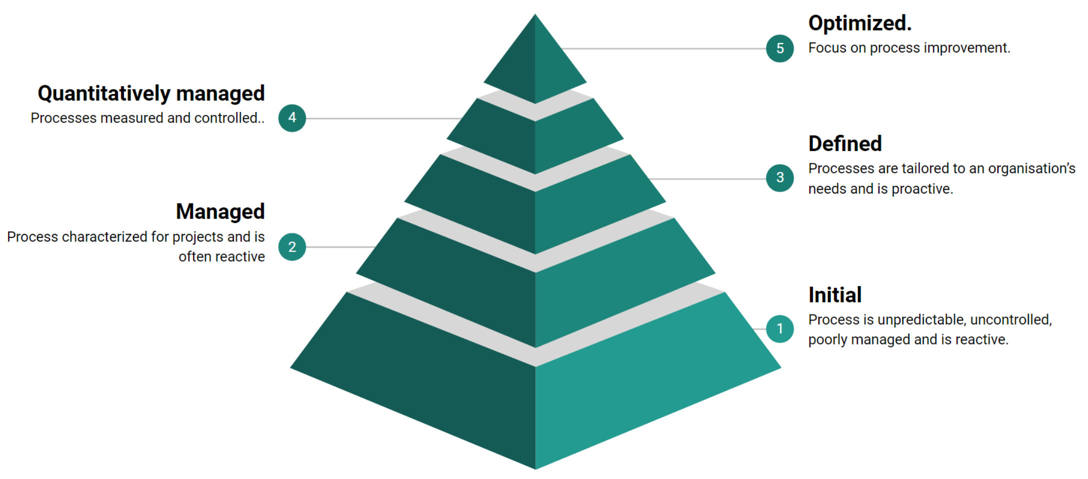

Optimized.
5
Focus on process improvement.
Quantitatively managed
Processes measured and controlled..
4
Defined
3
Processes are tailored to an organisation's needs and is proactive.
Managed
Process characterized for projects and is often reactive
2
Initial
1
Process is unpredictable, uncontrolled, poorly managed and is reactive.

Fig. 1 CMMI Levels

1.3.1 Capability Maturity Model level 3 and 4:

A capability level 3 process is characterized as a "defined process" A defined process is a managed (capability level 2) process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines, and contributes to work products, measures each task, and other properly defined information to the project knowledge.
A capability level 4 process is characterized as a "quantitatively managed process" A quantitatively managed process is a defined (capability level 3) process that is controlled using statistical and other quantitative techniques. Quantitative objectives for quality and process performance are established and used as criteria in managing the process. Quality and process performance is understood in statistical terms and is managed throughout the life of the process. level 4 demands Organizations at this level to meet the requirements of:
Data-driven approach: Use quantitative data to determine predictable processes that align with stakeholder needs
Measurable: Use defined metrics to demonstrate how processes benefit business operations
Refined: Repeatedly test, refine, and adapt processes in multiple conditions across the organization
Competent: Ensure all key stakeholders and process users are comfortable deploying the process in various environments and Adaptable Make it easy for processes to adapt to suit other

projects in the organization and serve as a template for future process development

Goals and plans:

Base quality goals and plans on the processes' proven ability to perform, and ensure they reflect contract requirements, to summarize it we can say:

CMMI Level 3: Defined

1. Process Standardization: Establish a set of standard processes for the organization.
2. Process Documentation: Document processes and procedures.
3. Training and Awareness: Provide training and awareness programs for employees.
4. Process Ownership: Assign process owners to oversee process implementation.
5. Continuous Improvement: Establish a culture of continuous improvement.
6. Requirements Management: Manage requirements throughout the project lifecycle.
7. Project Planning: Establish project plans and track progress.
8. Configuration Management: Manage changes to products and services.

CMMI Level 4: Quantitatively Managed

1. Quantitative Process Management: Establish quantitative objectives for process performance.
2. Process Performance Measurement: Collect and analyze data on process performance.
3. Statistical Process Control: Use statistical methods to control processes.
4. Predictive Modeling: Use predictive models to forecast process performance.
5. Innovation Management: Encourage innovation and experimentation.
6. Causal Analysis and Resolution: Identify and address root causes of problems.
7. Organizational Process Performance: Establish metrics to measure organizational process performance.
8. Quantitative Project Management: Use quantitative methods to manage projects.

To achieve CMMI Level 3, an organization must demonstrate a standardized and documented process framework, with a focus on continuous improvement. whereas to achieve CMMI Level 4, an organization must demonstrate quantitative management of processes, with a focus on data-driven decision making and predictive modeling.

Problem Statement:

Introduction:

A recent survey conducted to observe current trends in Software houses across the country revealed that most software houses are not implementing any process improvement model, furthermore, companies do not store knowledge gained from projects in a formal manner and do not maintain a project portfolio. This results in the absence of a standard process improvement model in the company.

Methodology:

I conducted a survey (using survey monkey webApp) across the country , targeting major cities Lahore Islamabad Rawalpindi and Karachi.. A total of 9 software houses were surveyed and (n=25) individuals participated. The roles ranged from project managers, CEOs , developers and testers
60% of these companies did 9 and above projects in the past two years.
This survey helped identify the current trends and areas of improvement in the software houses and their processes.
The two main questions that identified the problem were:
Q10. How much are you involved in different
SDLC phases?

Answer Choices	*Responses*
Very likely	66.67%
Somewhat likely	16.67%
Not sure	0.00%
Somewhat unlikely	0.00%
Very unlikely	16.67%

Q11. Do you formally follow any of the
well‑known process improvement models?

Answer Choices	*Responses*
Yes	83.33%
No	16.67%

Documentations are neglected overall as well, the data further reveals

Q27. What is the level of challenge you experience in
various software maintenance practices?
Answer Choices Responses
Lack of training 0.00%
Cost 0.00%
Complicated tech 16.67%
Outdated internal processes 0.00%
Ongoing support 16.67%
Lack of Traceability 0.00%
Lack of Code Comments 33.33%
Obsolete Legacy Systems 16.67%
Others ____________________ 16.67%

Q29. What other types of practices are carried out by your organization while supporting your software projects?
Answer Choices Responses
Determine Client Readiness 33.33%
Gain Commitment 0.00%
Align with Strategic Goals 16.67%
Company Culture 50.00%
Organization Structure 0.00%
Build a Solid Business Case 0.00%
Collect Feedback and Evaluate the Outcomes of The Change 0.00%
Other ____________________ 0.00%

Q31. What types of software project management
practices are carried out by your organization?
Answer Choices Responses
Collaboration 0.00%
Scheduling 0.00%
Issue Tracking 16.67%

Project Profile Management	0.00%
Scope Management	0.00%
Document Management	16.67%
Resource Management	0.00%
Estimation Management	0.00%
Project Risk Management	0.00%
All of the above	66.67%

The above question clearly shows project portfolio management is focused by just 66% of organizations.

Need and Importance of this research:

The research on developing an AI-based document parser for Software Requirements Specification (SRS) documents was necessary due to the limitations and challenges associated with traditional SRS document management. SRS documents are critical in software development, but they are often plagued by issues such as:

Ambiguity and unclear requirements, leading to misinterpretation and errors

Inconsistencies and incompleteness, causing delays and rework

Difficulty in maintaining and updating SRS documents, resulting in outdated or obsolete information. Limited collaboration and knowledge sharing among stakeholders, hindering effective communication and decision-making

The absence of automation and intelligence in SRS document management exacerbates these issues, making it essential to explore innovative solutions. This research addresses these challenges by introducing AI-driven metadata generation, enabling:

Improved clarity and precision in SRS documents

Enhanced analysis, validation, and testing capabilities

Facilitated collaboration and knowledge sharing among stakeholders

Increased efficiency and accuracy in SRS document management

By investigating the application of AI in SRS document parsing, this research aims to bridge the gap between traditional document management practices and the needs of modern software development, ultimately contributing to the creation of more robust, maintainable, and effective software systems.

Concluded Problem Statement:

We can summarize the problem statement as:

• *Most software houses currently do not implementing process improvement models and do not have any well defined process to Manage knowledge and systematically maintain a portfolio.*

Improper knowledge management hinders software development productivity.

Chapter 2

Literature Review:

For addressing this issue I reviewed literature from three different domains, this helped me curate a solution consisting of the knowledge gained from all three of these domains.

The concept of knowledge management has been widely discussed in the context of organizational success and failure. Davenport and Prusak(1998) [3] emphasize the importance of distinguishing between data, information, and knowledge, as these terms are often used interchangeably despite having distinct meanings. Data refers to discrete, objective facts about events, while information is a message that has the potential to change the way the receiver perceives something. Knowledge, on the other hand, is a fluid mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and information.

T. H. Davenport and L. Prusak's book, Working Knowledge: *How Organizations Manage What They Know*, provides a comprehensive examination of knowledge management within organizations. The authors argue that effective management of knowledge assets is crucial for organizational success, outlining strategies for capturing, sharing, and leveraging knowledge to enhance decision-making and innovation. This work is foundational for understanding how organizations can utilize their intellectual resources to gain competitive advantages and improve operational efficiency, including the development and optimization of technological solutions such as chatbots.

In a similar vein, A. K. Gupta and V. Govindarajan's [4] study on knowledge flows within multinational corporations explores how knowledge is disseminated and utilized across global operations. Their research sheds light on the challenges and strategies associated with managing knowledge in complex organizational structures. The findings are relevant for understanding how multinational companies can leverage global knowledge to drive innovation and improve practices, including those related to software and technology development. Their work highlights the importance of effective knowledge management in supporting global initiatives and technological advancements

Davenport and Prusak (1998)[3] argue that knowledge is not neat or simple, but rather a mixture of various elements that is fluid and formally structured. It is intuitive and therefore hard to capture in words or understand completely in logical terms. Knowledge exists within people and is often embedded in organizational routines, processes, practices, and norms. The authors also highlight the importance of understanding the characteristics of knowledge-intensive organizations and the need to evaluate knowledge by the decisions or actions to which it leads.

The distinction between data, information, and knowledge is crucial in the context of knowledge management initiatives. Many organizations fail to make a hard distinction between these terms in practice, and their initiatives often involve a mixture of knowledge and information, if not some data as well. Davenport and Prusak (1998) [3] observe that most organizations try to add value to what they have, moving it up the scale from data toward knowledge. However, this process requires humans to do virtually all the work, and the transformation of information into knowledge happens through activities such as comparison, consequences, connections, and conversation.

Overall, Davenport and Prusak's (1998) [3] work highlights the importance of understanding the nuances of knowledge management and the need to distinguish between data, information, and knowledge. Their framework provides a foundation for evaluating knowledge management initiatives and understanding the characteristics of knowledge-intensive organizations.

The researches done on CMMI includes work of W. S. Humphrey's [1] seminal work on the Capability Maturity Model (CMM) .It outlines a framework for improving software development processes by defining maturity levels that organizations can achieve. Humphrey emphasizes the importance of process discipline and continuous improvement, proposing a structured approach to enhancing software quality and performance. The model, detailed in "The Capability Maturity Model for Software," provides a roadmap for organizations to assess and improve their software development practices systematically. It is particularly relevant for understanding how structured process improvement can lead to better project outcomes and more reliable software systems.

Niazi, Wilson, and Zowghi's [14] framework for software process improvement focuses on designing effective strategies for implementing SPI initiatives. Their research identifies critical success factors and best practices for improving software development processes. The framework provides guidelines for addressing challenges and ensuring successful SPI adoption.

This work is valuable for enhancing software development practices, including those related to chatbot systems, by offering strategies for continuous improvement and process optimization.

Similarly M. C. Paulk's [2] exploration of Extreme Programming (XP) from a CMM perspective offers valuable insights into integrating agile methodologies with traditional process improvement models. Paulk examines how XP's iterative and flexible approach aligns with CMM's structured framework, suggesting that agile practices can complement and enhance process maturity efforts. This perspective is crucial for understanding how modern agile methodologies can be harmonized with established process improvement models to create effective and adaptable development processes. The discussion highlights the potential for blending different approaches to achieve superior software development outcomes.

R. G. Cooper's [5] article on portfolio management for new product development provides insights into managing and prioritizing innovation projects. Cooper's framework emphasizes the need for effective portfolio management to balance risk and return, allocate resources efficiently, and align projects with strategic objectives. This approach is particularly relevant for managing the development of new technologies, such as chatbots, where balancing innovation with practical implementation is critical. Cooper's work offers a structured methodology for evaluating and managing product development efforts to maximize their success and impact.

R. S. Kaplan and D. P. Norton's [6] work on the balanced scorecard introduces a strategic management tool that aligns organizational performance with strategic goals. Their approach focuses on using performance metrics to drive corporate synergies and ensure that all aspects of an organization are working towards common objectives. This framework is valuable for evaluating and enhancing the performance of technology projects, including chatbots, by providing a means to measure and manage various dimensions of success. Kaplan and Norton's balanced scorecard helps organizations ensure that their technological initiatives are aligned with overall business strategies and objectives.
A. M. Hickey and A. M. Davis [7] explore collaborative, goal-oriented approaches to requirements elicitation and validation in their study. They argue that effective requirements engineering is essential for the success of software projects and propose techniques for improving the accuracy and relevance of requirements. Their research highlights the importance of collaboration between stakeholders and the use of goal-oriented methods to ensure that requirements are well-defined and aligned with project objectives. This work is crucial for understanding how to effectively gather and validate requirements for chatbot systems and other

software applications.

Sommerville's later work, Software Engineering: A Comprehensive Approach [8], provides an updated overview of software engineering principles and practices. Sommerville addresses various aspects of software development, including lifecycle models, requirements engineering, and quality assurance. His comprehensive approach offers a broad understanding of software engineering methodologies and best practices, which are essential for developing and managing software projects, including chatbots. Sommerville's book is a key resource for grasping contemporary software engineering practices and their application in technological development.

Lalwani, Bhalotia, Pal, Bisen, and Rathod's [9] work on the implementation of a chatbot system using AI and NLP provides a practical exploration of chatbot development. Their study covers the integration of AI and NLP technologies to create functional and responsive chatbot systems. The authors discuss various technical aspects, including algorithms and frameworks used in building chatbots, and highlight the challenges faced during implementation. This research is valuable for understanding the practical considerations and technical requirements involved in developing effective chatbot systems.

Aslam's [10] study on the impact of artificial intelligence on chatbot technology offers an up-to-date analysis of advancements and innovations in the field. Aslam reviews the latest AI techniques, such as deep learning and transformer models, and their effects on improving chatbot capabilities. The study provides insights into how these technologies have enhanced chatbots' ability to understand and respond to user inputs more effectively. Aslam's research is essential for staying informed about current trends and future directions in AI-driven chatbot technology.

Lalwani, Bhalotia, Pal, Bisen, and Rathod's [9] study offers a detailed examination of the integration of AI and NLP technologies in chatbot development. While their work is valuable in outlining practical aspects and technical considerations, several *limitations* can be identified. Firstly, the study's focus on specific algorithms and frameworks may limit its applicability across different contexts. The effectiveness of the chatbot implementations described may vary depending on the specific requirements and constraints of other systems. This specificity can restrict the generalizability of their findings to broader or different chatbot applications. Secondly, while the paper highlights the challenges faced during implementation, it may not sufficiently address the scalability and adaptability of the proposed solutions. The study could benefit from a more thorough analysis of how the discussed technologies perform under varying loads and diverse user interactions. Scalability issues are particularly critical in real-world applications where chatbots need to handle high volumes of user queries efficiently. The study's practical insights are valuable, but a more in-depth exploration of scalability and long-term performance could enhance the utility of the research for developers seeking to deploy chatbots in dynamic environment. Aslam's [10] study provides a comprehensive overview of recent advancements in AI and their impact on chatbot technology, focusing on techniques like deep learning and transformer models. However, several shortcomings can be identified. Firstly, the paper's emphasis on cutting-edge AI techniques may overshadow practical implementation challenges. While the study provides valuable insights into the potential of new technologies, it may not fully address the complexities and resource requirements involved in integrating these techniques into functional chatbot systems. The gap between theoretical advancements and practical implementation can be a significant limitation for practitioners looking to apply these innovations. Secondly, Aslam's research might benefit from a more critical assessment of the limitations and ethical considerations associated with advanced AI techniques. For instance, deep learning models and transformers often require substantial computational resources, which can be a barrier for smaller organizations. Additionally, these models can sometimes produce biased or unpredictable outputs, raising concerns about fairness and reliability. A detailed of these issues would provide a more balanced view of the advancements and help stakeholders make more informed decisions regarding the adoption of these technologies.

In summary, while both studies provide significant contributions to the field of chatbot technology, they also have areas that require further exploration. Lalwani et al. [9] could enhance their work by addressing scalability and broader applicability, while Aslam's research would benefit from a more balanced view that includes practical implementation challenges and ethical considerations.

Abdellatif, Badran, Costa, and Shihab's comparative analysis of natural language understanding (NLU) platforms [11] for chatbots evaluates various tools based on their performance and usability. Their study provides a detailed assessment of NLU platforms, offering insights into their strengths and limitations. This comparative approach helps practitioners choose the most suitable NLU tools for their chatbot projects, based on criteria such as accuracy, scalability, and integration ease. The research is valuable for understanding the available options and making informed decisions about NLU platforms.

Another study involving professor Zowghi was reviewed. Gervasi, Ferrari, Zowghi, and Spoletini's [12] work on ambiguity in requirements engineering addresses a critical issue in software development. Their proposed framework aims to reduce ambiguity and improve the clarity of requirements, which is crucial for successful chatbot implementation. The authors argue that a unified approach to managing ambiguous requirements can enhance the quality and effectiveness of software systems. This research is relevant for improving requirements engineering practices and ensuring that chatbot systems meet user needs and project objectives.

Kassab, Ormandjieva, and Daneva's ontology-based approach [13] to non-functional requirements conceptualization provides a method for organizing and managing non-functional aspects of software systems. Their study emphasizes the importance of clearly defining requirements such as performance, security, and usability. By applying ontological frameworks, the authors offer a structured approach to understanding and addressing these requirements. This approach is particularly useful for developing robust chatbot systems that meet various non-functional criteria.

Jonatan and Aguilar-Alonso's research [15] on creating a chatbot based on NLP for WhatsApp addresses the unique challenges of deploying chatbots on popular messaging platforms. Their study explores the adaptation of NLP techniques to enhance user interaction and engagement on WhatsApp. The authors provide practical examples and implementation strategies for integrating chatbots into existing communication channels. This research is relevant for understanding how to effectively leverage chatbots in popular messaging environments.

Dr. Nalini, Karthik, and Koti's case study [16] on the Intellectual Interactive System offers insights into developing interactive systems using advanced technologies. Their work provides a practical example of how interactive systems can be designed and implemented, including considerations for user interaction and system functionality. The case study contributes to understanding the practical aspects of developing complex interactive systems, which can be applied to chatbot development.

Mane, Sonone, Gaikwad, and Ramteke's work [17] on smart personal assistants using machine learning explores the application of machine learning algorithms to enhance personal assistant capabilities. Their research highlights advancements in machine learning techniques and their impact on improving conversational agents. Although not exclusively focused on chatbots, their study provides valuable insights into how machine learning can be used to enhance the performance and functionality of chatbots.

Topsakal and Akinci’s [18] primer on creating large language model applications using Lang-chain offers a guide for developing LLM-based applications quickly and efficiently. Their research focuses on leveraging Lang-chain for rapid development of language model applications, including chatbots. The study provides practical insights into using advanced tools and techniques for developing sophisticated language-based applications. This research is useful for understanding how to streamline the development of LLM-based chatbots and other applications.

Davis, Dieste Tubio, Hickey, Juristo Juzgado, and Moreno’s study [19] on the effectiveness of requirements elicitation techniques provides empirical results from a systematic review. Their research evaluates various techniques for gathering and validating requirements, offering insights into their effectiveness in different contexts. This study is important for understanding how to improve requirements elicitation practices and ensure that software systems, including chatbots, meet user needs and expectations.

Lee and Choi’s work [20] on knowledge management enablers, processes, and organizational performance offers an integrative view of how knowledge management practices affect organizational outcomes. Their empirical examination highlights the role of knowledge management in enhancing performance and achieving strategic goals. This research is relevant for understanding how effective knowledge management can support the development and optimization of chatbot systems and other technological initiatives.

McAdam and McCreedy’s [21] critical review of knowledge management models examines various approaches and frameworks for managing organizational knowledge. Their analysis provides insights into the strengths and limitations of different models, contributing to a better understanding of knowledge management practices. This work is valuable for exploring how knowledge management can be applied to support technology development, including chatbots.

The ideas gathered from these literature reviews formed into a solution for the above mentioned problem.

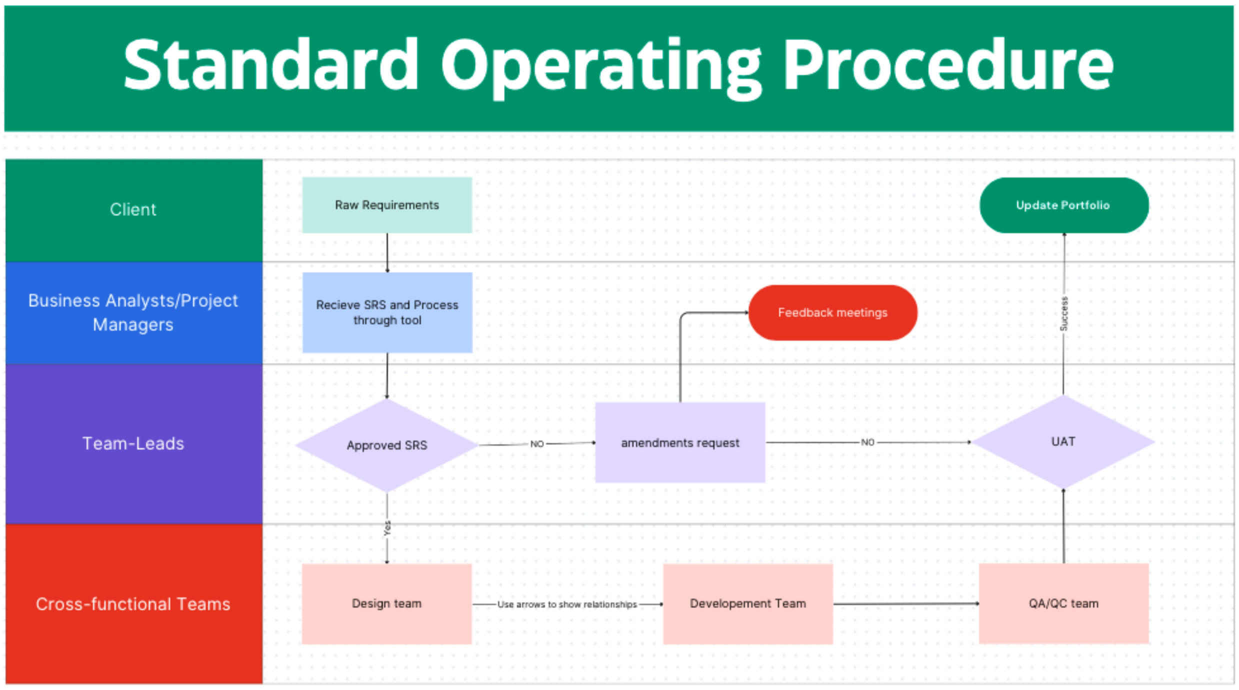
Chapter 3

Methodology:

The research method used in this thesis is Design Science Research (DSR). It's a methodology for developing and evaluating new artifacts in areas where this kind of results may help other researchers in their work. As per my survey data almost 17% of the companies do not formally follow any improvement models and 16% are not involved in all the formalities of SDLC. Here we need to keep in mind it is the major cities data and there are numerous startups in other cities throughout the country that are likely lagging behind on following process improvement models as well. Which highlights the importance of this research.

3.1 AI-based Requirement Gathering tool:

The implementation of a centralized Software Requirements Specification (SRS) document analysis tool yields numerous benefits for stakeholders involved in the software development process. By serving as a singular, authoritative source of truth, it ensures consistency and clarity in understanding project requirements, thereby mitigating the risk of misinterpretation and miscommunication. Furthermore, real-time updates and collaborative editing capabilities facilitate seamless information sharing and version control, enabling the tracking of changes and maintenance of a comprehensive revision history. Enhanced visibility and accessibility also empower stakeholders to make informed decisions and respond promptly to evolving project needs. Ultimately, a centralized SRS document contributes to increased transparency, accountability, and operational efficiency throughout the software development lifecycle.



The AI-based tool is able to receive a document, process it and then answer queries related to its content. The tool is able to extract relevant information and generate a response with accuracy, confidence and relevancy.

3.2 Implementation of Langchain

This study employed Langchain (an AI-driven approach) to enhance requirement gathering for portfolio and project management in these steps:

- Data Collection: Utilized AI technologies to document and analyze project requirements, scenarios, and user data.
- Scoring System: Developed a scoring system to measure accuracy, precision, and relevance of the gathered requirements, ensuring quantifiable and traceable results.
- Quantification and Documentation: Used AI to document and quantify requirements, scenarios, and user data, enabling a mature and transparent process.
- Evaluation and Validation: Assessed the effectiveness of the AI-driven approach using metrics such as accuracy, precision, and relevance scores and drew parallels with CMMI level requirements.

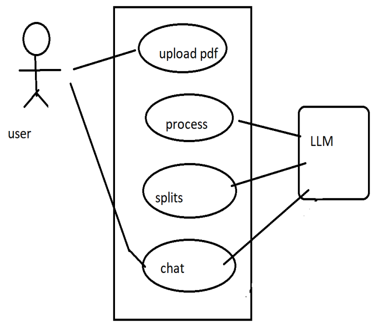
3.2.1 How it works:

1. User Input: User asks a specific question related to a document.
2. Document Retrieval: LLM retrieves the relevant document or a corpus of documents.
3. Contextual Understanding: LLM analyzes the document content, identifying relevant sections and paragraphs.
4. Information Extraction: LLM extracts specific information related to the user's query.
5. Response Generation: LLM generates a concise response, summarizing the extracted information.

Use-Case diagram:

Below is a use **case diagram** of my system.

Fig. 2 Use case diagram



3.3 Brief description of Code Components and Functionality:

- Library Imports and Environment Setup: The code begins by importing necessary libraries such as os, openai, and panel. It also loads environment variables from a .env file using load_dotenv(find_dotenv()) and sets up the OpenAI API key.
- PDF Processing Function: The load_db function processes a PDF file by splitting it into text chunks, creating embeddings, and setting up a question-answering system.
- ChatBot Class: This class manages the chatbot behavior, including storing chat history, processing queries, and generating responses. It also includes methods for uploading and processing PDF files, detecting count queries, counting elements, estimating confidence, and sending queries. OpenAI API key is a must have to operate this.
- User Interface Components: The code defines various interactive elements, such as file input, text input, buttons, and text output displays, to facilitate user interaction.
- Event Handling and Callbacks: The code includes functions to update the chatbot response, handle Excel file downloads, and manage user input.
- Panel Layout and Styling: The user interface is organized into tabs (Chat, Metadata, Confidence Scores & Accuracy) with custom CSS styling applied to the tabs.

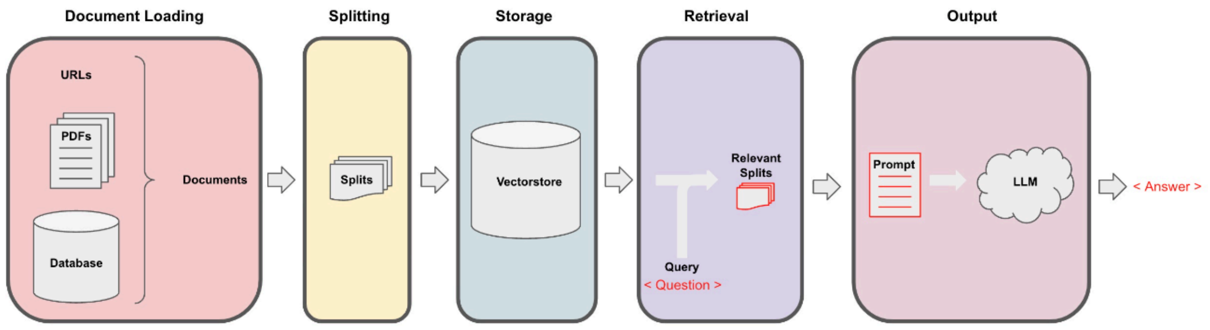


Fig:3 Work-flow diagram

You can give system prompts like:

- "Extract all user stories from this SRS document"
- "How many user types are mentioned in this SRS document?"
- "What are the roles and responsibilities of each user type?"

- "How many actors are mentioned in this SRS document?"
- "Extract all functional requirements related to user authentication"

Fulfilling CMMI level 3 requirements:

As per ***CMMI level 3*** requirements, data should be structured and tailored according to an organization's needs. For this we can use the AI capability of categorizing and extracting relevant information from raw data. Lang-chain and other LLM models can be fine-tuned in a lot of different ways, making it possible to formulate your own criteria for data extraction
for example: Lang-chain uses the following parameters to categorize a requirement as a functional requirement:

1. Verb: The requirement statement starts with a verb like "shall", "will", "must", "should", or "may".
2. Action-oriented: The requirement describes an action or behavior that the system or product must perform.
3. Specific: The requirement is clear, concise, and unambiguous.
4. Measurable: The requirement can be measured or tested to determine if it's met.
5. Product-focused: The requirement describes a characteristic or behavior of the product or system.
6. Non-technical: The requirement doesn't describe technical details or implementation-specific information.
7. User-centric: The requirement is written from the user's perspective, describing what they need or want.
8. Functional phrase: The requirement contains phrases like "The system shall", "The product will", or "The user can".
9. Input/Output: The requirement describes an input or output of the system or product.
10. System behavior: The requirement describes how the system or product responds to a particular situation or input.

By analyzing these parameters, Lang-chain can determine if a requirement is functional requirement or not, meaning it describes what the system or product should do, rather than how it should do it (which would be a non-functional requirement). similarly other prompts/queries can be customized by setting parameters for the selected model as per an organization's needs. Additionally meta data containing time-stamp, page number are added to make the extracted information as traceable as possible.

Fulfilling CMMI level 4 Requirements:

Whereas ***CMMI level 4*** is satisfied by quantifying the extracted information using scoring function.

In Lang-chain, relevance score, confidence score, and other similar metrics are used to evaluate the quality and accuracy of responses generated by language models. Here's a brief explanation of each:

Relevance Score measures how well the response aligns with the input prompt or question. It indicates the degree to which the response addresses the topic or task at hand.

The relevance score calculation in Lang-chain typically involves a combination of natural language processing (NLP) techniques and machine learning algorithms. Below are general outline of the steps:

1. Text Embeddings: Convert the input prompt and response into numerical representations (embeddings). These embeddings capture semantic meaning and context.
2. Similarity Measurement: Calculate the similarity between the input prompt and response embeddings using metrics like:
3. Cosine similarity and Dot product

Apply a Scoring Function to the similarity measurement, often using a weighted sum or a neural network. This function maps the similarity value to a relevance score, usually between 0 and 1.

Normalization: Normalize the relevance score to ensure consistency across different input prompts and responses.

Some common relevance scoring functions include:

a. Simple similarity:

$$Relevance = similarity(p, r)$$

, where p is the prompt and r is the response.

b. Weighted sum:

$$Relevance = w1 * similarity(p, r) + w2 * keyword\_match(p, r)$$

where w1 and w2 are weights.

4. Confidence Score: It represents the model's certainty or confidence in its response. A higher confidence score suggests the model is more likely to be correct.

Confidence Score (CS): A numerical value between 0 and 1, indicating the model's confidence in its response.

Calculation:

a. Probability Distribution: LangChain's Large Language Model (LLM) generates a probability

distribution over the possible output tokens. This distribution represents the model's uncertainty about the next token.
b. Top-K Tokens: Select the top-K tokens with the highest probabilities from the distribution. K is a hyperparameter, typically set to 5 or 10.
c. Cumulative Probability: Calculate the cumulative probability of the top-K tokens. This represents the model's confidence in the selected tokens.

5. Confidence Score: Normalize the cumulative probability by dividing it by the sum of probabilities of all tokens in the vocabulary. This ensures the confidence score is bounded between 0 and 1.

Mathematical Representation:

$$CS = (\sum[\text{top-K probabilities}]) / (\sum[\text{all token probabilities}])$$

Explanation:
CS → 1: High confidence, the model is certain about its response.
CS → 0: Low confidence, the model is uncertain or exploring multiple possibilities.
Accuracy Score: Assesses the correctness or accuracy of the response, often requiring human evaluation or external validation.
These scores are calculated using various techniques, such as
1) Post-processing algorithms i.e. Analyze the response's content, syntax, and semantics.
2) Model-internal metrics: Leverage the language model's internal workings, like attention weights or probability distributions as well as knowledge graphs.
Our AI-based tool allows us to view these scores for the extracted information and export it in excel format for storing purposes.

3.4 Table to draw parallels between CMMI levels and corresponding AI tool capabilities:

AI capability:	CMMI level	Contribution in KM
Faster and easier extraction of requirements using natural language processing (NLP) and text generation	Defined (3)	Resource management, cost management, time management.
Detection/ resolution of ambiguities, inconsistencies, and redundancies in the requirements Identification and prioritization of the most important and relevant requirements	Quantitatively managed Controlled (4)	Repeatable and traceable
Verification and validation of the requirements against the business goals and user needs	Defined (3)	Measurable Data driven decisions
Traceability impact analysis of requirements across the project artifacts	Managed and Controlled (4)	Quality of customer satisfaction Measured progress
Collaboration communication among stakeholders and development team	Controlled and Optimizing (4 -> 5)	Resource management/time management.

Table.1 (4.1) Parallels between CMMI level and AI capabilities

Chapter 4

Results and Discussion

The above mentioned methodology achieved the working product as well as successfully achieved CMMI (Capability Maturity Model Integration) Level 3 and 4 by demonstrating a well-defined, quantifiable, and repeatable process for requirement gathering and project knowledge management.

The integration of AI-based semantic annotation in Software Requirements Specification (SRS) documents brings numerous benefits and significance. By adding context and meaning to requirements, semantic annotation improves clarity and precision, reducing ambiguity and ensuring stakeholders share a common understanding. This, in turn, enables enhanced analysis and validation, facilitated collaboration and knowledge sharing, increased efficiency and accuracy, better traceability, and improved reusability and maintainability.
Ultimately, AI-based semantic annotation represents a significant advancement in software engineering, transforming the way we create, manage, and utilize SRS documents. By harnessing AI's potential, organizations can enhance the quality and reliability of software systems, improve collaboration and knowledge sharing, increase efficiency and accuracy, reduce errors, costs, and time-to-market, and unlock the full potential of their SRS documents, leading to better software development outcomes and improved business results.

The solution was verified and validated on PUblicly available REquirements documents. The **PURE** [23] is a repository of **79 Software Requirements Specifications (SRS)** of different types. 63 SRSs are pdf, 14 of them are word documents, and 3 of them are HTML. We used only pdf files.

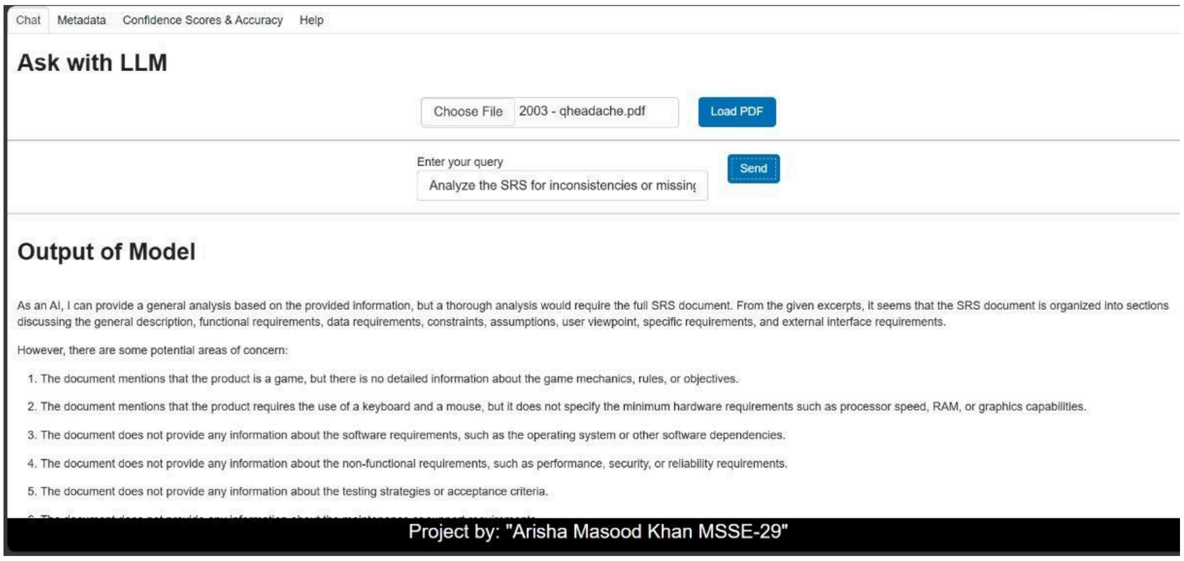
4.1 LLM Model's Results:

The LLM model was able to successfully extract information and answer queries with an average **confidence score of 8 and median accuracy was 7**. The time-stamps and document's metadata was extracted, stored and exported with a 100% accuracy.
Calculated based on Entity Recognition Questions where n=63 PDF from PURE dataset.
we got the following results:

Tool	**TP**	**FP**	**TN**	**FN**
GPT4	29	7	18	9
Langchain with GPT4	54	8	21	10

RAG Tool	**Precision Score**	**F1 Score**	**Recall**
GPT4 alone	0.806	0.784	0.763
Langchain + GPT4	0.859	0.845	0.831

We can observe the results from the following screen-shots



Ask with LLM

2003 - qheadache.pdf

Enter your query

Analyze the SRS for inconsistencies or missing

Output of Model

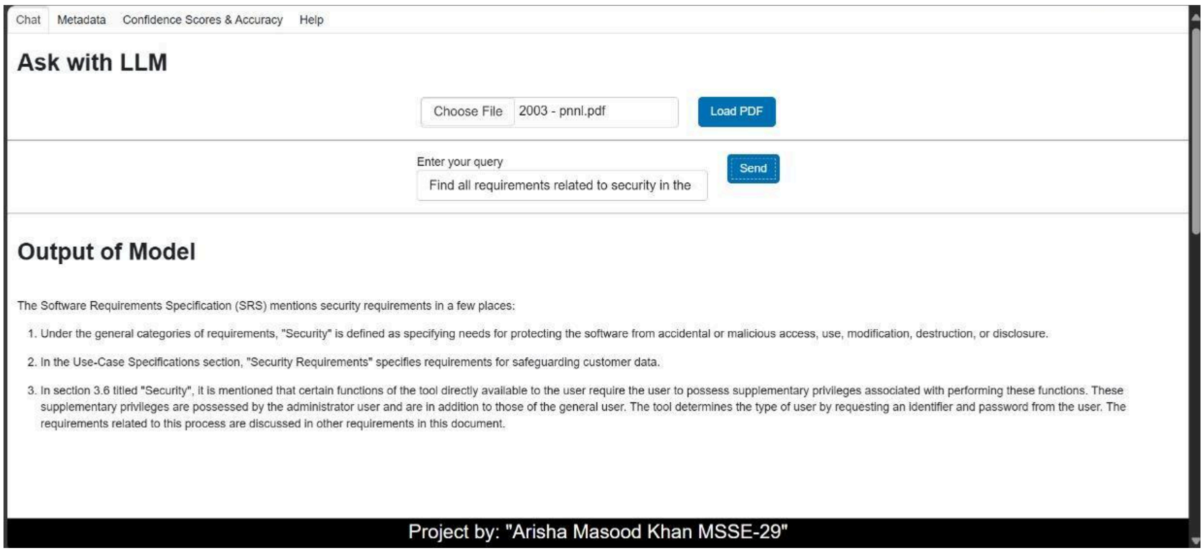
As an AI, I can provide a general analysis based on the provided information, but a thorough analysis would require the full SRS document. From the given excerpts, it seems that the SRS document is organized into sections discussing the general description, functional requirements, data requirements, constraints, assumptions, user viewpoint, specific requirements, and external interface requirements.

However, there are some potential areas of concern:

1. The document mentions that the product is a game, but there is no detailed information about the game mechanics, rules, or objectives.
2. The document mentions that the product requires the use of a keyboard and a mouse, but it does not specify the minimum hardware requirements such as processor speed, RAM, or graphics capabilities.
3. The document does not provide any information about the software requirements, such as the operating system or other software dependencies.
4. The document does not provide any information about the non-functional requirements, such as performance, security, or reliability requirements.
5. The document does not provide any information about the testing strategies or acceptance criteria.

Project by: "Arisha Masood Khan MSSE-29"

Image 1



Ask with LLM

2003 - pnnl.pdf

Enter your query

Find all requirements related to security in the

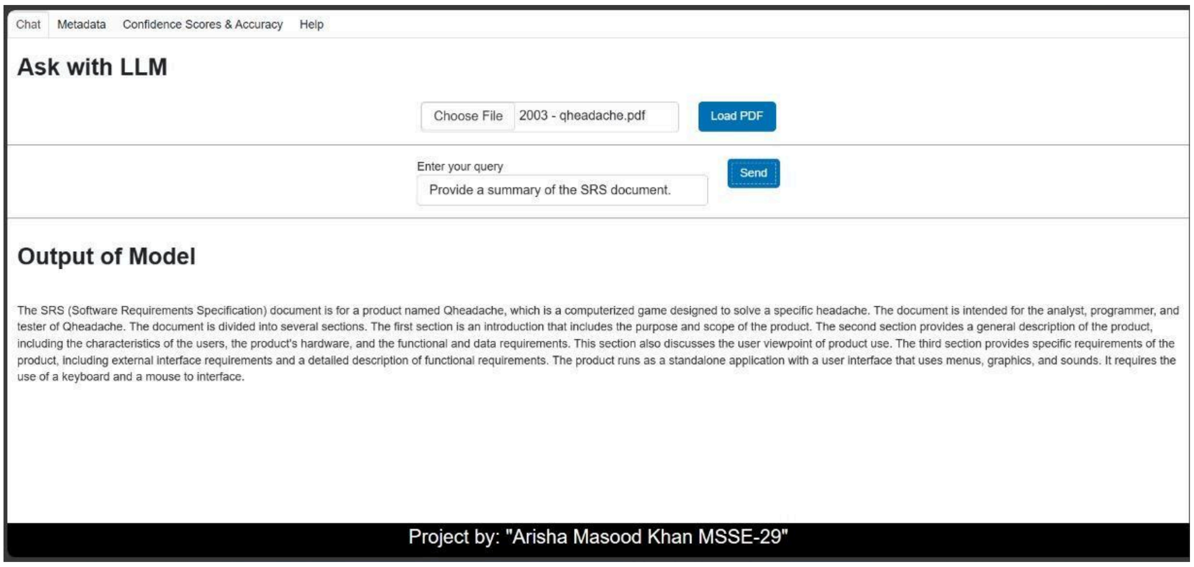
Output of Model

The Software Requirements Specification (SRS) mentions security requirements in a few places:

1. Under the general categories of requirements, "Security" is defined as specifying needs for protecting the software from accidental or malicious access, use, modification, destruction, or disclosure.
2. In the Use-Case Specifications section, "Security Requirements" specifies requirements for safeguarding customer data.
3. In section 3.6 titled "Security", it is mentioned that certain functions of the tool directly available to the user require the user to possess supplementary privileges associated with performing these functions. These supplementary privileges are possessed by the administrator user and are in addition to those of the general user. The tool determines the type of user by requesting an identifier and password from the user. The requirements related to this process are discussed in other requirements in this document.

Project by: "Arisha Masood Khan MSSE-29"

Image 2



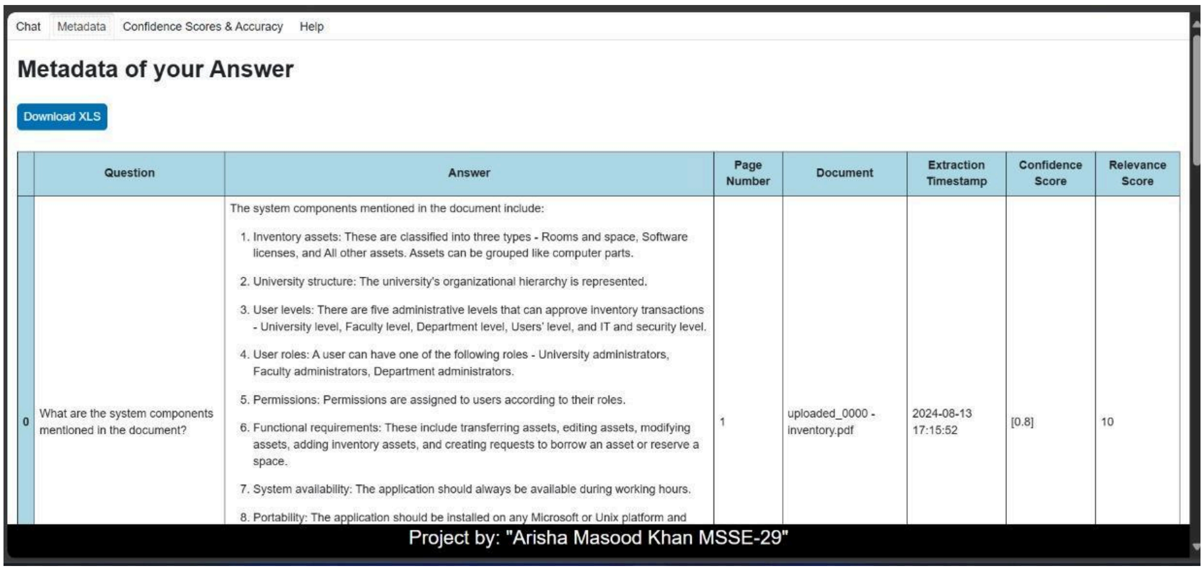
Ask with LLM

Output of Model

The SRS (Software Requirements Specification) document is for a product named Qheadache, which is a computerized game designed to solve a specific headache. The document is intended for the analyst, programmer, and tester of Qheadache. The document is divided into several sections. The first section is an introduction that includes the purpose and scope of the product. The second section provides a general description of the product, including the characteristics of the users, the product's hardware, and the functional and data requirements. This section also discusses the user viewpoint of product use. The third section provides specific requirements of the product, including external interface requirements and a detailed description of functional requirements. The product runs as a standalone application with a user interface that uses menus, graphics, and sounds. It requires the use of a keyboard and a mouse to interface.

Project by: "Arisha Masood Khan MSSE-29"

Image 3

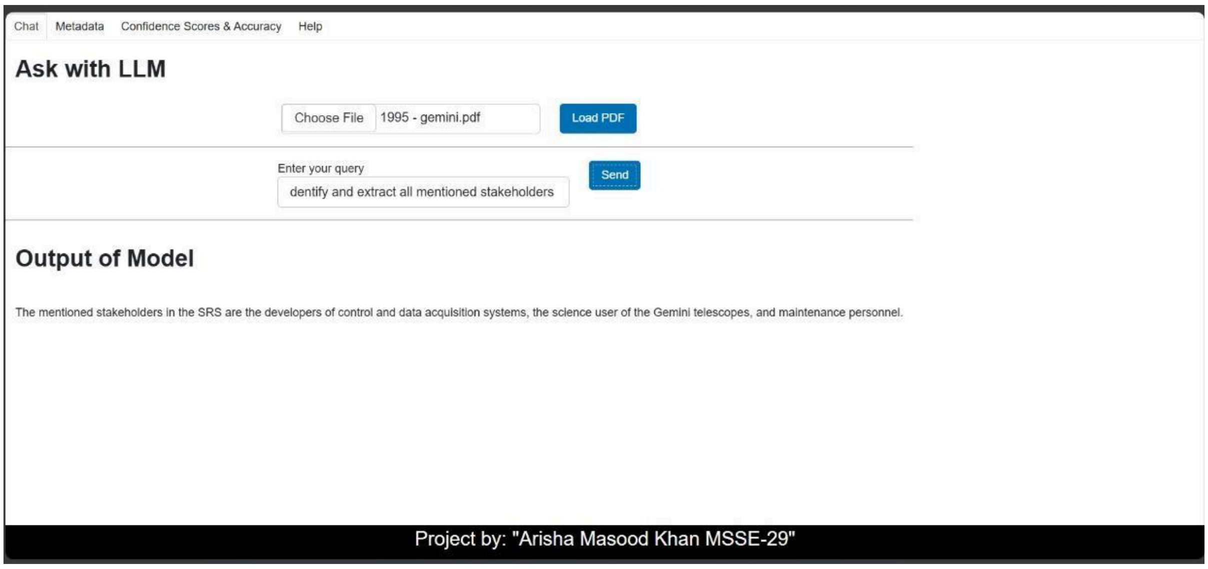


Metadata of your Answer

	Question	Answer	Page Number	Document	Extraction Timestamp	Confidence Score	Relevance Score
0	What are the system components mentioned in the document?	The system components mentioned in the document include: 1. Inventory assets: These are classified into three types - Rooms and space, Software licenses, and All other assets. Assets can be grouped like computer parts. 2. University structure: The university's organizational hierarchy is represented. 3. User levels: There are five administrative levels that can approve inventory transactions - University level, Faculty level, Department level, Users' level, and IT and security level. 4. User roles: A user can have one of the following roles - University administrators, Faculty administrators, Department administrators. 5. Permissions: Permissions are assigned to users according to their roles. 6. Functional requirements: These include transferring assets, editing assets, modifying assets, adding inventory assets, and creating requests to borrow an asset or reserve a space. 7. System availability: The application should always be available during working hours. 8. Portability: The application should be installed on any Microsoft or Unix platform and	1	uploaded_0000 - inventory.pdf	2024-08-13 17:15:52	[0.8]	10

Project by: "Arisha Masood Khan MSSE-29"

Image 4


Chat
Metadata
Confidence Scores & Accuracy
Help

Ask with LLM

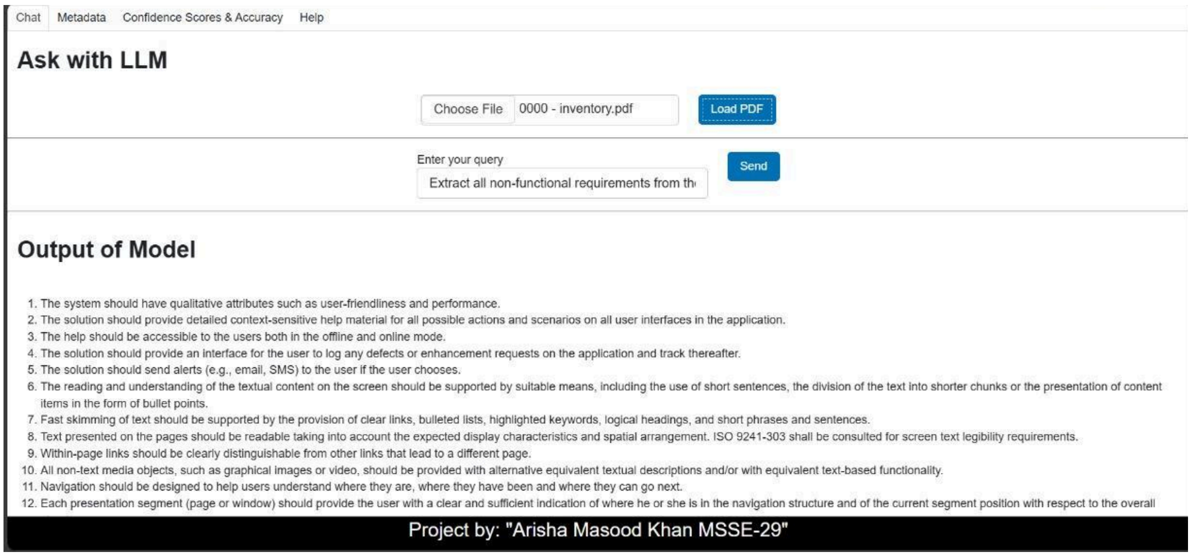
Choose File
1995 - gemini.pdf
Load PDF
Enter your query
dentify and extract all mentioned stakeholders
Send

Output of Model

The mentioned stakeholders in the SRS are the developers of control and data acquisition systems, the science user of the Gemini telescopes, and maintenance personnel.

Project by: "Arisha Masood Khan MSSE-29"

Image 5

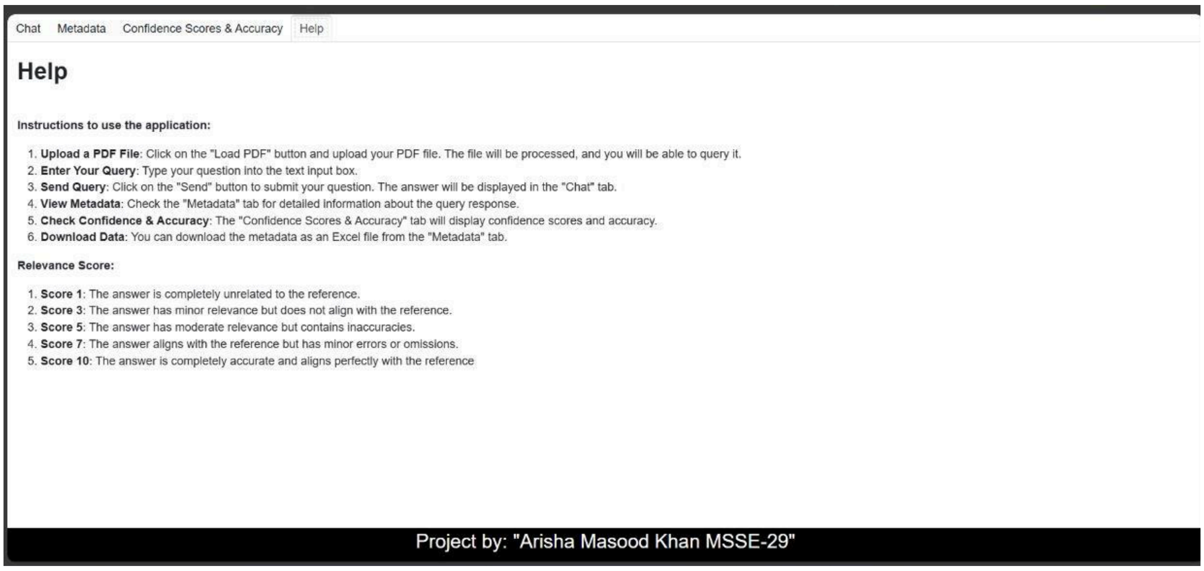

Chat
Metadata
Confidence Scores & Accuracy
Help
Choose File
0000 - inventory.pdf
Load PDF
Enter your query
Extract all non-functional requirements from th
Send
Project by: "Arisha Masood Khan MSSE-29"

Ask with LLM

Output of Model

1. The system should have qualitative attributes such as user-friendliness and performance.
2. The solution should provide detailed context-sensitive help material for all possible actions and scenarios on all user interfaces in the application.
3. The help should be accessible to the users both in the offline and online mode.
4. The solution should provide an interface for the user to log any defects or enhancement requests on the application and track thereafter.
5. The solution should send alerts (e.g., email, SMS) to the user if the user chooses.
6. The reading and understanding of the textual content on the screen should be supported by suitable means, including the use of short sentences, the division of the text into shorter chunks or the presentation of content items in the form of bullet points.
7. Fast skimming of text should be supported by the provision of clear links, bulleted lists, highlighted keywords, logical headings, and short phrases and sentences.
8. Text presented on the pages should be readable taking into account the expected display characteristics and spatial arrangement. ISO 9241-303 shall be consulted for screen text legibility requirements.
9. Within-page links should be clearly distinguishable from other links that lead to a different page.
10. All non-text media objects, such as graphical images or video, should be provided with alternative equivalent textual descriptions and/or with equivalent text-based functionality.
11. Navigation should be designed to help users understand where they are, where they have been and where they can go next.
12. Each presentation segment (page or window) should provide the user with a clear and sufficient indication of where he or she is in the navigation structure and of the current segment position with respect to the overall

Image 6


Chat
Metadata
Confidence Scores & Accuracy
Help
Project by: "Arisha Masood Khan MSSE-29"

Help

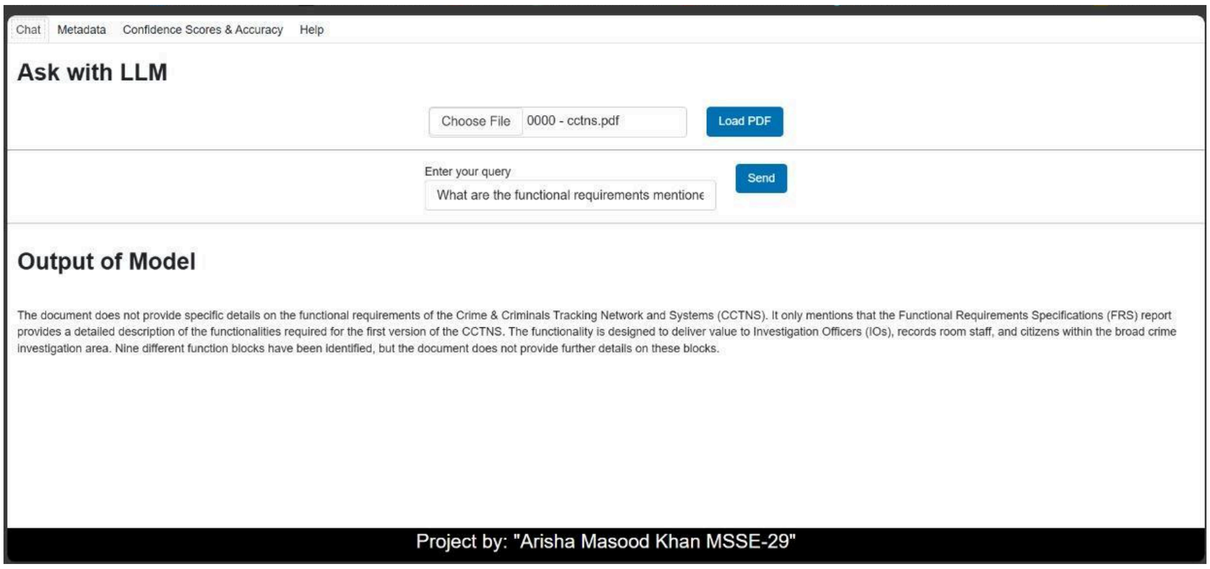
Instructions to use the application:

1. **Upload a PDF File**: Click on the "Load PDF" button and upload your PDF file. The file will be processed, and you will be able to query it.
2. **Enter Your Query**: Type your question into the text input box.
3. **Send Query**: Click on the "Send" button to submit your question. The answer will be displayed in the "Chat" tab.
4. **View Metadata**: Check the "Metadata" tab for detailed information about the query response.
5. **Check Confidence & Accuracy**: The "Confidence Scores & Accuracy" tab will display confidence scores and accuracy.
6. **Download Data**: You can download the metadata as an Excel file from the "Metadata" tab.

Relevance Score:

1. **Score 1**: The answer is completely unrelated to the reference.
2. **Score 3**: The answer has minor relevance but does not align with the reference.
3. **Score 5**: The answer has moderate relevance but contains inaccuracies.
4. **Score 7**: The answer aligns with the reference but has minor errors or omissions.
5. **Score 10**: The answer is completely accurate and aligns perfectly with the reference

Image 7


Chat
Metadata
Confidence Scores & Accuracy
Help
Choose File
0000 - cctns.pdf
Load PDF
Enter your query
What are the functional requirements mentione
Send
Project by: "Arisha Masood Khan MSSE-29"

Ask with LLM

Output of Model

The document does not provide specific details on the functional requirements of the Crime & Criminals Tracking Network and Systems (CCTNS). It only mentions that the Functional Requirements Specifications (FRS) report provides a detailed description of the functionalities required for the first version of the CCTNS. The functionality is designed to deliver value to Investigation Officers (IOs), records room staff, and citizens within the broad crime investigation area. Nine different function blocks have been identified, but the document does not provide further details on these blocks.

The system efficiently answered a variety of PDF related queries along with it's meta-data, including but not limited to:

1. Content extraction:

What are the functional requirements mentioned in the SRS?

Extract all non-functional requirements from the document.

2. Specific requirement queries:

What is the requirement ID XYZ mentioned in the SRS?

What is the description of requirement ABC?

3. Entity recognition:

Identify and extract all mentioned stakeholders in the SRS.

What are the system components mentioned in the document?

4. Relationship queries:

What are the dependencies between requirements XYZ and ABC?

Which requirements are related to system component DEF?

5. Summarization:

Provide a summary of the SRS document.

Summarize the functional requirements mentioned in the SRS.

6. Search and filter:

Find all requirements related to security in the SRS.

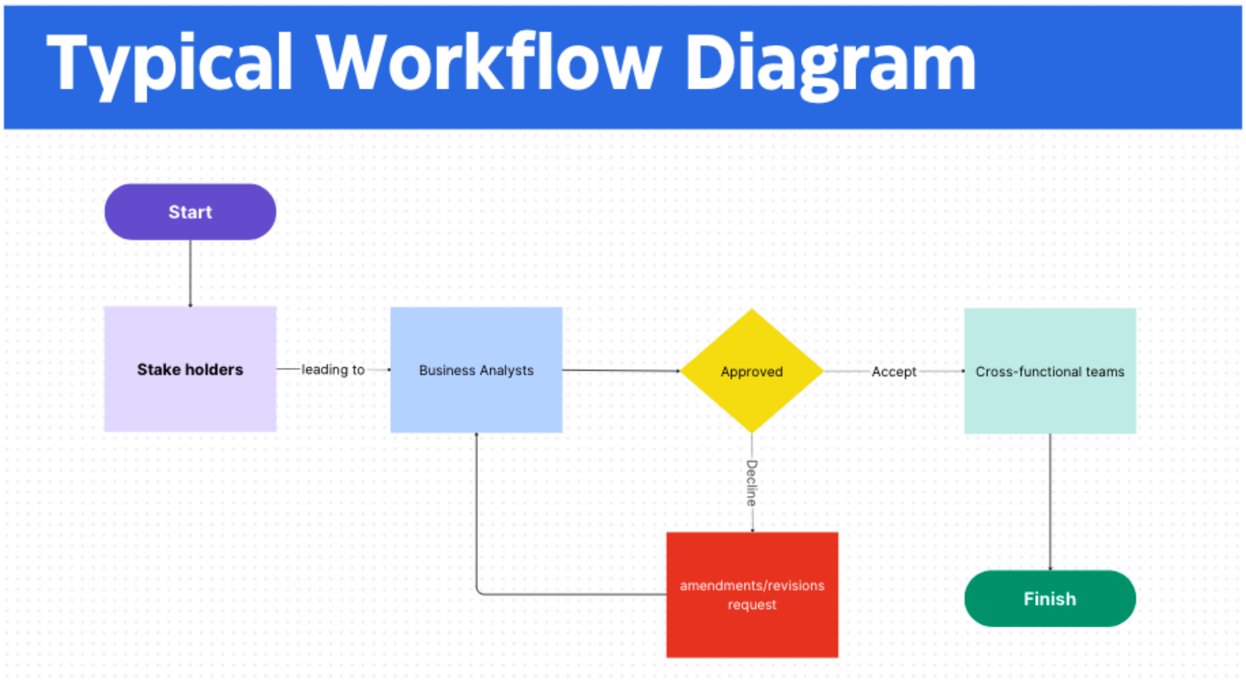
Filter requirements by priority level (high, medium, low).

4.2 Comparison of CMMI level requirements to AI-tools Capabilities

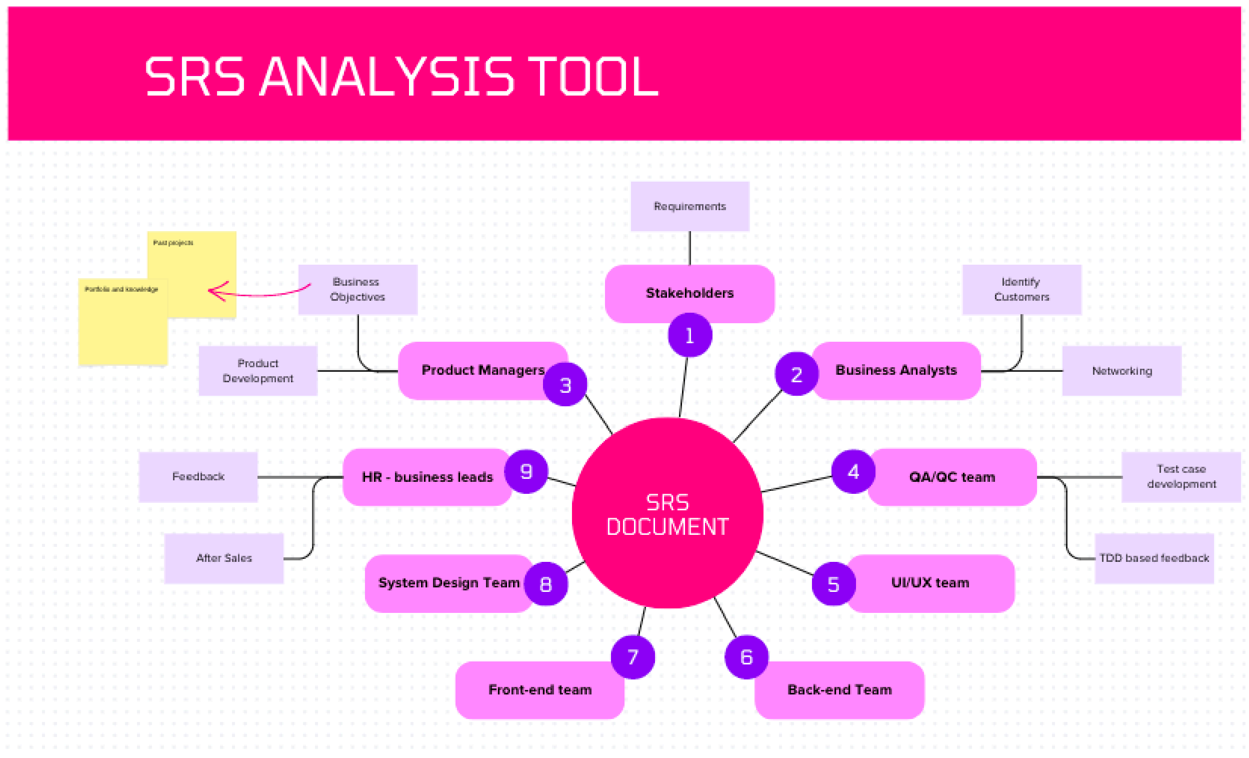
CMMI level 3	AI-tool feature	CMMI level 4	AI-tool feature
Standardized	➔ User is able to define Parameters and fine-tune model	*Statistical Process Control*	➔ User is able to keep track of extracted information through Key performance indicators
Documented	➔ User is able to export extracted information	*Managed and Controlled Process*	➔ Data visualization from the extracted information helps manage project
Process Ownership:	➔ Knowledge sharing through extracted information	*Quantitative Process Performance Management*	➔ LLM model's capability to count and keep scores of extracted data ensures Quantitative management.
Requirements Management:	➔ Changes are tracked through version controlling and time stamps	*Organizational Process Performance and analysis*	➔ Extracted information provides instant work break down helping in team analysis and performance
Project Planning:	➔ Data visualization from the extracted information helps manage project	*Predictive Modeling*	➔ Effective portfolio management from well documented projects data results in generating accurate predictions

4.3 Analysis and Discussion of Results obtained:

In a software house, the current workflow of SRS (Software Requirements Specification) knowledge typically begins with the receipt of a project request from a client or stakeholder. The project manager and business analysts then collaborate to gather and document the initial requirements through meetings, interviews, and surveys. The gathered information is then compiled into a draft SRS document, which is reviewed and refined by the development team, quality assurance team, and other relevant stakeholders. Once the SRS document is finalized, it serves as a guide for the development team to create the software, and for the quality assurance team to develop testing plans. Throughout the development process, the SRS document is continuously updated and refined to reflect any changes or new requirements that arise. Finally, the completed software is validated against the SRS document to ensure that it meets all the specified requirements.



A centralized SRS document analysis tool provides a single source of truth for all stakeholders, ensuring everyone is on the same page regarding project requirements. It facilitates real-time updates and collaboration, reducing misunderstandings and miscommunications. This centralized repository also enables version control, tracking changes, and maintaining a history of updates. Moreover, it allows for easy access and visibility, enabling stakeholders to make informed decisions and take prompt actions. Overall, a centralized SRS document enhances transparency, accountability, and efficiency throughout the software development lifecycle.



The results yielded from extracting knowledge in a standardized way can be utilized by:

1. Setting Quantitative Goals:

Companies can establish specific, measurable project objectives, such as:

- Schedule performance index (SPI)
- Cost performance index (CPI)

Schedule Performance Index (SPI) and Cost Performance Index (CPI)

Schedule Performance Index (SPI):

Measures project progress in relation to the schedule.

SPI = Earned Value (EV) / Planned Value (PV)

1 indicates ahead of schedule, < 1 indicates behind schedule.

Cost Performance Index (CPI):

Measures project expenses in relation to the budget.

CPI = Earned Value (EV) / Actual Cost (AC)

1 indicates under budget, < 1 indicates over budget.

2. Customer satisfaction ratings

Feedback and the meeting between stake holders can be analysed using accuracy, relevance scores and sentiment analysis on the documented communication.

3. Easy and fast Earned Value Management (EVM) calculations:

Use EVM to track project progress, comparing planned value (PV) to earned value (EV) and actual cost (AC). This can also be done by generating graphs and charts from the extracted data.

4. Statistical Process Control (SPC):

Apply SPC techniques to monitor and control project processes, such as:
SWOT analysis and duration schedules.

5. Process capability analysis through:

- Control charts
- Statistical quality control

6. Quantitative Risk Management:

Identify, assess, and prioritize risks using quantitative methods, such as:

- Monte Carlo simulations - (technique that uses random sampling to solve complex problems and predict outcomes. It involves generating multiple scenarios based on probability distributions and analyzing the results to estimate potential outcomes. This method is commonly used in finance, engineering, and other fields to manage risk and make informed decisions.)
- Threat analysis
- Expected monetary value (EMV) analysis

7. Performance Metrics and Indicators:

Track and analyze project performance using metrics such as:

- Burn-down/burn-up charts
- Cycle time
- Lead time
- Throughput

8. Data-Driven Decision Making:

Use quantitative data to inform project decisions, such as:

- Resource allocation
- Schedule adjustments
- Budget reallocation
- Change requests

9. Predictive Analytics:

Apply statistical models and machine learning algorithms to extract and forecast project outcomes, such as:

- Project duration
- Cost overruns

- Quality issues

10. Quantitative Stakeholder Management:

Use data to communicate project performance to stakeholders, including:

- Progress reports
- Performance dashboards
- Data visualizations - excel export facilitates data visualization

Chapter 5
Conclusion

5.1 Introduction:

Maintaining a portfolio of projects is essential for software houses and organizations to effectively manage their projects and resources. A systematically maintained portfolio provides numerous benefits, including improved project visibility, resource allocation, risk management, and decision-making. Storing this knowledge also contributes to applying lessons learned from one project to another.It is essential for continuous improvement and effective project management.

By applying these quantitative techniques, project managers can make data-driven decisions, optimize project performance, and achieve better outcomes.

Knowledge management and project portfolio management are therefore complementary and mutually beneficial practices that can lead to greater organizational success and process improvement. The advanced AI-tools we can gather requirements keeping in mind the ultimate goal of knowledge management and portfolio management in such a way so as to satisfy the CMMI level 3 and 4 demands. This will help reach a process maturity level which can benefit the company's overall success and managerial processes. From this research it can be concluded that software houses need to incorporate AI from the very beginning of the project, so it can help track all the related data till the very end of the project and that knowledge can be used later on as well, because it'll be systematically stored in the portfolio for future reference.

5.2 Achieved Research Goals:

To achieve our objective we document insights, successes, and challenges encountered during the project by drawing conclusions from our knowledge base and portfolio.

I. Improved Project Visibility:

A systematically maintained portfolio allows organizations to have a comprehensive view of all ongoing and upcoming projects. This visibility enables better resource allocation, risk management, and decision-making at the organizational level.

II. Effective Resource Allocation:

By maintaining a portfolio of projects, organizations can allocate resources more effectively based on project priorities, timelines, and resource availability. This ensures optimal utilization of resources and minimizes bottlenecks in project execution.

III. **Enhanced Risk Management:**

Systematically maintained portfolios facilitate better risk management by identifying and mitigating risks across multiple projects. Organizations can prioritize risk mitigation strategies and allocate resources accordingly to minimize potential project disruptions.

IV. **Informed Decision-Making:**

A well-maintained portfolio provides decision-makers with relevant information and insights to make informed decisions about project investments, resource allocations, and strategic priorities. This helps organizations align their project portfolios with their overall business objectives and maximize return on investment.

V. **Continuous Improvement:**

Maintaining a portfolio allows organizations to track project performance, identify areas for improvement, and implement lessons learned from past projects. This continuous improvement cycle enables organizations to enhance their project management practices and achieve better outcomes over time which is also corresponds to last level of CMMI.

5.3 Limitations:

The limitations of the proposed solution are:

- In-house resistance to adoption of the system. When a new system is introduced there is usually resistance from the team using old system as they have become accustomed to it.
- The system doesn't accommodate run-time change requests, making it less flexible but more reliable.
- AI is unable to read-between-the-lines or comprehend the undertone to a document. This system will not be able to comprehend the common sense based expectations of stakeholders that humans can intercept.

5.4 Future Work:

While AI-based tools have shown promising advancements in supporting requirements gathering, there are opportunities for further research and development.This research has far-reaching implications for software engineering, paving the way for further innovations in AI-driven SRS document management.
Future research directions include:

- Integrating different AI-based tools with existing requirements management systems in different organizations.

- Developing more sophisticated techniques for extracted requirements analysis apart from relevance score etc.

- Further research can be done in developing customized NLP parameters and AI chat-bots fine-tuning for an industry's unique needs.

- Investigating the use of machine learning algorithms for knowledge sharing within an organization by making parts of knowledge available to different teams as per their needs and permissions.
- expansion to other document types and fields of study.

By presenting this research, I contribute to the advancement of software engineering, demonstrating the potential of AI to transform the way we develop and maintain software systems.

5.5 Conclusion:

From this research it can be concluded that software houses need to incorporate AI from the very beginning of the project, so it can help track all the related things till the very end of the project and that knowledge can be used later on as well, because it'll be systematically stored in the portfolio for future reference. In conclusion, the application of semantic annotation in Software Requirements Specification (SRS) documents has far-reaching benefits that transform the software development process. By infusing requirements with meaning and context, semantic annotation enhances clarity, precision, and accuracy, leading to a more reliable and effective SRS document. The advantages of semantic annotation extend beyond the development phase, supporting ongoing maintenance, evolution, and reuse of software systems.

Through semantic annotation, we can unlock the full potential of SRS documents, enabling automated analysis, validation, and testing, while fostering collaboration, knowledge sharing, and informed decision-making. As software systems continue to grow in complexity, the importance of semantic annotation in SRS documents will only continue to increase, serving as a foundation for building robust, maintainable, and adaptable software systems.

Ultimately, the strategic adoption of semantic annotation in SRS documents represents a significant step forward in software engineering, poised to revolutionize the way we develop, maintain, and evolve software systems. By embracing this innovative approach, we can create software systems that are more responsive to user needs, more resilient to change, and more aligned with business objectives.

Chapter 6

Contribution and Areas of Application

The research on "AI-based requirements gathering to improve knowledge management and portfolio management" ultimately improves SDLC process.

6.1 My Research Contribution

This research presents a novel approach to enhancing Software Requirements Specification (SRS) documents through the development of an AI-based document parser. Leveraging the capabilities of LangChain and OpenAI, this parser automates the process of adding metadata to SRS documents, revolutionizing the way we create, manage, and utilize these critical documents.

6.1.1 Key Contributions:

A. Innovative Application of AI:

This research demonstrates the first application of LangChain and OpenAI in SRS document parsing, showcasing the potential of AI in software engineering.

B. Automated Metadata Generation:

The developed parser automatically adds relevant metadata to SRS documents, enhancing their clarity, precision, and accuracy.

C. Improved SRS Document Quality:

By infusing SRS documents with metadata, this research enables better analysis, validation, and testing, ultimately leading to more robust and maintainable software systems.

D. Enhanced Collaboration and Knowledge Sharing:

The model facilitates collaboration among stakeholders by providing a shared understanding of SRS documents, fostering knowledge sharing and informed decision-making.

6.2 Areas of Application:

It can have various areas of application, each contributing to enhancing organizational processes and outcomes. Some areas of application include:

6.2.1 Software Development Lifecycle Optimization:

Implementing AI-based requirements gathering can streamline the process of identifying, documenting, and managing requirements throughout the software development lifecycle. This optimization can lead to improved efficiency, reduced development time, and enhanced quality of software products.

6.2.2 Knowledge Management Enhancement:

By leveraging AI techniques for requirements gathering, organizations can better capture, organize, and share knowledge related to project requirements. This can lead to a more effective portfolio management system, enabling easy access to relevant information and facilitating collaboration among team members.

6.2.3 Portfolio Management Optimization:

AI-based requirements gathering can assist in selecting and prioritizing projects within a portfolio based on their alignment with organizational goals and strategic objectives. This optimization can lead to better resource allocation, improved risk management, and increased ROI for the organization's project portfolio.

6.2.4 Process Model Improvement (CMMI):

Integrating AI-based requirements gathering with established process models like CMMI can enhance the maturity and effectiveness of organizational processes. By leveraging AI-driven insights and automation, organizations can identify areas for process improvement, implement best practices, and achieve higher levels of process maturity as per the CMMI framework.

6.2.5 Risk Management and Mitigation:

AI-powered analysis of requirements data can help identify potential risks and dependencies early in the project lifecycle. By integrating this analysis into portfolio management processes, organizations can proactively mitigate risks, optimize resource allocation, and improve project success rates.

6.2.6 Decision Support Systems:

AI-based requirements gathering can feed into decision support systems that assist stakeholders in making informed decisions related to project selection, resource allocation, and process improvement initiatives. These systems can leverage AI-driven analytics to provide actionable insights and recommendations for optimizing organizational processes and achieving strategic objectives.

6.2.7 Continuous Improvement and Learning:

By continuously analyzing requirements data using AI techniques, organizations can identify patterns, trends, and areas for improvement in their knowledge management, portfolio management, and process models. This fosters a culture of continuous improvement and learning, enabling organizations to adapt to changing market dynamics and technological advancements effectively.

6.2.8 Cross-Functional Collaboration:

AI-based requirements gathering can facilitate cross-functional collaboration by providing a centralized platform for stakeholders from different departments to contribute, review, and prioritize requirements. This collaborative approach enhances communication, alignment, and coordination across various organizational functions, ultimately improving overall project outcomes.

6.3 Summary:

The research on AI-based requirements gathering has broad applications across knowledge management, portfolio management, and process model improvement, offering opportunities for organizations to enhance efficiency, effectiveness, and strategic alignment in their software development and project management practices.

Semantic annotation in SRS documents enhances the clarity and precision of requirements, enabling stakeholders to comprehend the intended functionality and constraints of the software system. By annotating requirements with relevant context and meaning, ambiguities are reduced, and misinterpretations are minimized, resulting in a more accurate and reliable SRS document.

Semantic annotation also facilitates the analysis and validation of requirements, enabling automated tools to check for consistency, completeness, and correctness. This leads to the early detection of errors, inconsistencies, and gaps in the requirements, reducing the risk of downstream defects and rework. Furthermore, annotated requirements provide a foundation for automated testing, traceability, and impact analysis, streamlining the software development process.

The benefits of AI based semantic annotation in SRS documents extend beyond the development phase, as they also support ongoing maintenance, evolution, and reuse of software systems. Annotated requirements serve as a knowledge base, capturing the rationale, constraints, and assumptions underlying the system's design and functionality. This enables developers, maintainers, and future stakeholders to understand the system's context and make informed decisions, reducing the risk of introducing unintended changes or defects.

Bibliography

[1] W. S. Humphrey, ”The Capability Maturity Model for Software,” Journal
of Systems and Software, vol. 59, no. 2, pp. 151-163, 2002.
[2] M. C. Paulk, ”Extreme programming from a CMM perspective,” International Journal of Software Engineering and Knowledge Engineering,
vol. 11, no. 1, pp. 19-36, 2001.
[3] T. H. Davenport and L. Prusak, Working knowledge: How organisations
manage what they know. Harvard Business Press, 1998.
[4] A. K. Gupta and V. Govindarajan, ”Knowledge flows within multinational corporations,” International Journal of Information Management,
vol. 20, no. 3, pp. 169-186, 2000.
[5] R. G. Cooper, ”Portfolio management for new product development,”
Project Management Journal, vol. 32, no. 2, pp. 4-14, 2001.
[6] R. S. Kaplan and D. P. Norton, ”Alignment: Using the balanced
scorecard to create corporate synergies,” International Journal of Project
Management, vol. 24, no. 2, pp. 137-149, 2006.
[7] A. M. Hickey and A. M. Davis, ”Requirements elicitation and validation
using a collaborative, goal-oriented approach,” Requirements Engineering Journal, vol. 8, no. 2, pp. 131-144, 2003.
[8] I. Sommerville, ”Software engineering: A comprehensive approach,”
International Journal of Software Engineering and Knowledge Engineering, vol. 16, no. 5, pp. 651-674, 2006.
[9] T. Lalwani, S. Bhalotia, A. Pal, S. Bisen, and V. S. Rathod, ”Implementation of a Chat Bot System using AI and NLP,” in Proceedings of the
[Conference Name], 2018, pp. 1-5.
[10] F. Aslam, ”The Impact of Artificial Intelligence on Chatbot Technology:
A Study on the Current Advancements and Leading Innovations,”
European Journal of Technology, vol. 2023, pp. 1-12, 2023.
[11] A. Abdellatif, K. M. Badran, D. E. Costa, and E. Shihab, ”A Comparison
of Natural Language Understanding Platforms for Chatbots in Software
Engineering,” IEEE Transactions on Software Engineering, vol. 48, pp.
3087-3102, 2020.

[12] V. Gervasi, A. Ferrari, D. Zowghi, and P. Spoletini, ”Ambiguity in Requirements Engineering: Towards a Unifying Framework,” in From Software Engineering to Formal Methods and Tools, and Back, 2019, pp. 1-15.
[13] M. Kassab, O. Ormandjieva, and M. Daneva, ”An Ontology Based Approach to Non-functional Requirements Conceptualization,” in 2009 Fourth International Conference on Software Engineering Advances, 2009, pp. 299-308.
[14] M. Niazi, D. Wilson, and D. Zowghi, ”A framework for assisting the design of effective software process improvement implementation strategies,” Journal of Systems and Software, vol. 78, no. 2, pp. 204-222, 2005.
[15] V. Jonatan and I. Aguilar-Alonso, ”Creation Of A ChatBot Based On Natural Language Processing For Whatsapp,” ArXiv, vol. abs/2310.10675, 2023.
[16] N. Dr. Nalini, K. Karthik, and N. R. Koti, ”Intellectual Interactive System (a case study of NMIT),” in Proceedings of the [Conference Name], 2022, pp. 1-5.
[17] P. Mane, S. Sonone, N. Gaikwad, and J. Ramteke, ”Smart personal assistant using machine learning,” in 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 368-371.
[18] O. Topsakal and T. C. Akinci, ”Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast,” International Conference on Applied Engineering and Natural Sciences, 2023.
[19] A. M. Davis, O. Dieste Tubio, A. M. Hickey, N. Juristo Juzgado, and A. M. Moreno, ”Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review,” in 14th IEEE International Requirements Engineering Conference (RE’06), 2006, pp. 179-188.
[20] Lee, Heeseok and Choi, Byounggu. (2003). Knowledge Management Enablers, Processes, and Organizational Performance: An Integrative View and Empirical Examination.J. of Management Information Systems. 20. 179-228.

10.1080/07421222.2003.11045756.
[21] McAdam, R. and McCreedy, S. (1999), ”A critical review of knowledge management models”, The Learning Organization, Vol. 6 No. 3, pp. 91-101
[22] Killen, Catherine Jugdev, Kam Drouin, Nathalie Petit, Yvan. (2012). Advancing Project and Portfolio Management Research: Applying Strategic Management Theories. International Journal of Project Management. 30. 525–538. 10.1016/j.ijproman.2011.12.004
[23] Ferrari, Alessio & Spagnolo, Giorgio & Gnesi, Stefania. (2017). PURE: A Dataset of Public Requirements Documents. 502-505. 10.1109/RE.2017.29.