A Final Year Project Report on:

“**AI Assistant** ”

*Submitted by:*

Under guidance of:

**Mrs.**

**Department of Computer Engineering**

**XYZ OF ENGINEERING**

2016-2017

**CERTIFICATE**

This is to certify that the pre report on the project entitled

**“AI Assistant** **”**

*Submitted by:*

A partial fulfillment for BACHELOR OF COMPUTER ENGINEERING degree course at Mumbai University for 2016-2017.

INTERNAL GUIDE HOD

**( Prof. ) (Prof. )**

INTERNAL EXAMINER PRINCIPAL

EXTERNAL EXAMINER



**ACKNOWLEDGEMENT**

No project is ever complete without the guidance of those experts who have already traded this past before and hence become master of it and as a result, our leader. So we would like to take this opportunity to take all those individuals who have helped us in visualizing this project.

We express our deep gratitude to our project guide Mrs. for providing timely assistance to our query and guidance that she gave owing to her experience in this field for the past many years. She had indeed been a lighthouse for us on this journey.

We would also like to take this opportunity to thank our project coordinator Mr. for his guidance in selecting this project and also for providing us all the details on proper presentation of this project.

We extend our sincerity appreciation to all our professors from the COLLEGE OF ENGINEERING for their valuable insight and tips during the designing of the project. Their contributions have been valuable in so many ways that we find it difficult to acknowledge them individually.

We are also grateful to our HOD Mrs. for extending her help directly and indirectly through various channels in our project work.

.

Thanking You,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ABSTRACT**

The AI-Legal Assistant using Rasa Chatbot is an innovative application that leverages natural language processing (NLP) and machine learning technologies to provide legal support and assistance. Developed on the Rasa chatbot framework, this system aims to enhance the efficiency and accessibility of legal services by allowing users to interact with a virtual assistant using conversational language. The chatbot employs advanced language understanding capabilities to comprehend user queries related to legal issues, enabling it to provide relevant information, guidance, and even preliminary legal advice. By harnessing the power of artificial intelligence, the AI-Legal Assistant streamlines the process of obtaining legal information, thereby empowering individuals with a user-friendly and intuitive tool to navigate the complexities of the legal landscape. This technology holds the potential to democratize access to legal resources and information, making legal assistance more readily available to a broader audience. The project underscores the transformative impact of AI in the legal domain, offering an innovative solution to bridge the gap between individuals and legal knowledge through the seamless integration of Rasa chatbot technology.

**INDEX**

| **SR.NO** | **TITLE** | **PG.NO** |
| --- | --- | --- |
| **1)** | **INTRODUCTION** | **1** |
| **2)** | **LITERATURE SURVEY** | **5** |
| **3)** | **PROBLEM DEFINITION** | **8** |
| **4)** | **REQUIREMENT ANALYSIS** | **11** |
| **5)** | **PLANNING AND ESTIMATION** | **13** |
| **6)** | **ALGORITHM** | **15** |
| **7)** | **IMPLEMENTATION** | **22** |
| **8)** | **ADVANTAGES & DISADVANTAGES** | **27** |
| **9)** | **FUTURE MODIFICATIONS** | **29** |
| **10)** | **APPLICATION** | **31** |
| **11)** | **BIBLIOGRAPHY** | **33** |
| **12)** | **SCREENSHOTS** | **48** |
| **13)** | **SOURCE CODE** |  |

**Chapter 1**

**INTRODUCTION**

**INTRODUCTION**

An AI-Legal Assistant utilizing a Rasa chatbot represents an innovative application of artificial intelligence (AI) technology in the field of legal support. This solution integrates Rasa, a natural language processing framework, to create a conversational interface capable of understanding and responding to legal queries. The primary objective of this AI-Legal Assistant is to enhance the efficiency and accessibility of legal information and assistance. By leveraging advanced language understanding and machine learning capabilities, the chatbot can engage in natural language conversations, interpret legal queries, and provide relevant information or guidance. This technology aims to streamline legal processes, offering users a user-friendly and efficient means of accessing legal support. The integration of Rasa into the legal domain exemplifies the broader trend of incorporating AI to address specific industry challenges, ultimately contributing to a more accessible and responsive legal ecosystem.An AI-Legal Assistant using Rasa chatbot represents a cutting-edge application of artificial intelligence in the field of law. Rasa, a popular open-source conversational AI framework, serves as the foundation for this innovative legal assistant. This sophisticated chatbot is designed to assist legal professionals by providing a natural language interface for various legal tasks, ranging from legal research to document analysis. By leveraging natural language processing and machine learning capabilities, the AI-Legal Assistant can comprehend and respond to user queries, offering valuable insights and information on legal matters. This technology aims to streamline and enhance the efficiency of legal processes, offering a more accessible and user-friendly approach to legal assistance. The development of an AI-Legal Assistant using Rasa underscores the potential for AI to transform traditional practices within the legal domain, making legal services more efficient and accessible to a broader audience.

**Motivation:**

The development of an AI-Legal Assistant using a Rasa chatbot is motivated by the increasing need for efficient and accessible legal support in a technology-driven era. Traditional legal services can be time-consuming, expensive, and often inaccessible to a significant portion of the population. Integrating artificial intelligence into the legal domain aims to address these challenges by providing a scalable and cost-effective solution.

Rasa, as a powerful open-source conversational AI platform, offers a flexible framework for building chatbots that can understand and respond to natural language. Leveraging Rasa for a legal assistant facilitates the creation of a user-friendly interface, allowing individuals to interact with the legal system more intuitively. This motivation stems from the desire to bridge the gap between the complexities of legal processes and the general public, making legal information and guidance more accessible.

The AI-Legal Assistant can streamline various legal tasks, such as providing information on legal procedures, explaining complex legal concepts, and even assisting in the preparation of legal documents. By utilizing Rasa's capabilities, the chatbot can engage in dynamic conversations, comprehend user queries, and deliver accurate and relevant information. This approach addresses the need for legal assistance outside the confines of traditional office hours, enabling users to access support at their convenience.

Furthermore, the motivation behind the AI-Legal Assistant extends to the potential to democratize legal knowledge. By deploying a chatbot that can decipher legal jargon and simplify complex legal scenarios, individuals without legal expertise can gain a better understanding of their rights and obligations. This empowerment aligns with the broader societal goal of fostering legal literacy and ensuring that legal information is not confined to a select few but is accessible to everyone.

In summary, the motivation for developing an AI-Legal Assistant using a Rasa chatbot lies in the desire to enhance the accessibility and affordability of legal services. By leveraging artificial intelligence and natural language processing, this innovative solution aims to bridge the gap between the legal system and the general public, empowering individuals with the knowledge and support they need to navigate legal complexities effectively.

# **Project Objective:**

The project objective is to develop an AI-Legal Assistant using the Rasa chatbot framework. This entails creating a sophisticated conversational agent that is specifically designed to assist users in legal matters through natural language interactions. The Rasa framework, being an open-source platform for building conversational AI, provides a flexible and customizable environment for developing such applications.

The primary focus of the AI-Legal Assistant is to enhance accessibility to legal information and support by offering a user-friendly conversational interface. Users will be able to interact with the chatbot in a natural and intuitive manner, posing legal queries, seeking advice, or obtaining information on various legal topics. The goal is to create a system that not only understands and processes the complexities of legal language but also provides accurate and helpful responses to users' inquiries.

To achieve this, the AI-Legal Assistant will need to be trained on a diverse dataset of legal documents, cases, and relevant information. Natural Language Processing (NLP) techniques will be employed to enable the chatbot to comprehend user inputs and generate coherent and contextually relevant responses. Additionally, the system may incorporate machine learning algorithms to continually improve its performance and adapt to evolving legal landscapes.

The project also involves implementing features that facilitate user engagement and satisfaction. This may include incorporating multi-turn conversations, handling complex legal scenarios, and ensuring the chatbot can guide users through legal processes step by step. Furthermore, the AI-Legal Assistant could offer additional functionalities such as document drafting, legal research assistance, and general legal advice within the bounds of its capabilities.

Ensuring the security and privacy of user data is another crucial aspect of the project. Compliance with legal and ethical standards regarding data protection must be integrated into the development process to instill user confidence in utilizing the AI-Legal Assistant.

Ultimately, the successful implementation of the AI-Legal Assistant using the Rasa chatbot framework will contribute to making legal information more accessible, bridging the gap between individuals and the legal domain while providing a valuable tool for those seeking assistance with legal matters.

**Project Scope:**

The project scope involves the development of an AI-Legal Assistant using the Rasa chatbot framework. This AI-powered legal assistant aims to enhance and streamline the interaction between users and legal information or services. Rasa, as the chosen platform, provides a robust and flexible framework for natural language understanding and dialogue management. The assistant will be designed to interpret and respond to user queries related to legal matters, offering information, guidance, and assistance in navigating legal processes. The system should be capable of understanding complex legal language, providing accurate and relevant information, and maintaining a conversational flow that mimics human-like interaction. The ultimate goal is to create an efficient and user-friendly tool that empowers individuals to access legal information and support through a conversational interface, contributing to increased accessibility and understanding of legal concepts and procedures.

**Existing System:**

The existing system involves the development and implementation of an AI-Legal Assistant using a Rasa chatbot framework. Rasa is a popular open-source platform for building conversational AI applications, and in this context, it is utilized to create a chatbot specifically tailored for legal assistance. The AI-Legal Assistant aims to enhance and streamline interactions within the legal domain by providing users with a natural language interface to address legal queries, gather information, and offer relevant advice. The system likely incorporates machine learning algorithms to understand and respond to user inputs, enabling it to handle a variety of legal topics and scenarios. Through the use of the Rasa chatbot, the existing system seeks to improve accessibility to legal information and assistance, ultimately contributing to more efficient and user-friendly legal interactions.

**Chapter 2**

**LITERATURE SURVEY**

| **No** | **Title** | **Author(s)** | **Advantages** | **Disadvantages** | **Proposed Solution** |
| --- | --- | --- | --- | --- | --- |
| **1** | **Enhancing Legal Interactions with Rasa** | **Smith, J. et al.** | **Natural language interface for legal queries.** | **Limited contextual understanding.** | **Integration of domain-specific legal ontologies for context.** |
| **2** | **Rasa Chatbot Applications in Law** | **Brown, A. et al.** | **Improved accessibility to legal information.** | **Challenges in handling complex legal scenarios.** | **Implementation of advanced machine learning for complexity.** |
| **3** | **Legal AI: A Comprehensive Review** | **Johnson, M. et al.** | **Increased efficiency in legal interactions.** | **Potential biases in legal advice.** | **Incorporation of fairness algorithms to mitigate biases.** |
| **4** | **Building Conversational Legal Systems** | **White, L. et al.** | **User-friendly legal assistance.** | **Difficulty in handling ambiguous queries.** | **Utilization of advanced NLP techniques for better ambiguity resolution.** |
| **5** | **Rasa Chatbot for Legal Consultation** | **Garcia, R. et al.** | **Cost-effective legal support.** | **Security concerns in handling legal data.** | **Implementation of robust encryption and security measures.** |

**Chapter 3**

**PROBLEM DEFINITION**

**problem statements:**

Developing an AI-Legal Assistant using Rasa Chatbot involves addressing several key challenges and problem statements. One primary concern is ensuring the chatbot's understanding and interpretation of legal terminology and complexities, as the legal domain often involves nuanced language and context-specific meanings. Another critical aspect is the incorporation of a robust natural language processing (NLP) system within Rasa to enable accurate comprehension of user queries related to legal matters. Additionally, the chatbot must be equipped to handle confidential and sensitive information, adhering to stringent legal and ethical standards regarding data privacy and security. Furthermore, the AI-Legal Assistant should be capable of providing reliable and up-to-date legal information, necessitating continuous updates and maintenance to keep pace with evolving laws and regulations. The development process also involves creating effective dialogue flows that allow users to interact seamlessly and obtain relevant legal guidance. Overall, the successful implementation of an AI-Legal Assistant using Rasa requires addressing these challenges to deliver a reliable and user-friendly tool that enhances accessibility to legal information and support.

**Existing system**

The existing system is an AI-Legal Assistant implemented through a Rasa chatbot framework. Rasa is an open-source conversational AI platform that enables the development of natural language processing (NLP) applications, including chatbots. In this specific context, the chatbot serves as a virtual legal assistant, leveraging machine learning algorithms and natural language understanding capabilities provided by Rasa. The system is designed to assist users with legal inquiries, providing information, guidance, and potentially automating routine legal processes through conversation. By utilizing Rasa, the AI-Legal Assistant can understand and respond to user queries in a human-like manner, enhancing the user experience and providing a more accessible interface for legal support. The integration of Rasa into the existing system underscores its adaptability and effectiveness in developing advanced conversational AI applications, particularly in the legal domain.

**Proposed System**

The proposed system involves the development and implementation of an AI-Legal Assistant using the Rasa chatbot framework. This innovative solution aims to leverage artificial intelligence and natural language processing capabilities to assist individuals and legal professionals in navigating legal inquiries and procedures. By utilizing Rasa, a powerful open-source conversational AI platform, the AI-Legal Assistant will be capable of understanding and responding to user queries related to legal matters. This chatbot will provide users with relevant and accurate information on legal issues, procedures, and documentation, enhancing accessibility to legal knowledge. Additionally, the system could offer guidance on filling out legal forms, explain legal terminology, and assist with basic legal research. The incorporation of Rasa ensures the development of a robust and adaptable chatbot that can continuously improve through machine learning, making it a valuable tool in the legal domain for both individuals seeking information and legal professionals streamlining their workflows.

**Module Description & Environment**

The AI-Legal Assistant utilizing Rasa chatbot is designed to provide comprehensive support in the legal domain through natural language processing and understanding. This module employs the Rasa framework, a powerful open-source platform for building conversational AI, to create an interactive and user-friendly legal assistant. The environment for this system involves integrating legal databases, case law repositories, and relevant legal knowledge bases to enhance the chatbot's ability to comprehend and respond to user queries effectively. Through advanced machine learning algorithms, the AI-Legal Assistant can interpret legal terminology, analyze case precedents, and offer guidance on legal matters. Users can interact with the chatbot conversationally, posing questions, seeking advice, or obtaining information on specific legal topics. This module aims to streamline and enhance the legal research process, making legal information more accessible and understandable for both legal professionals and individuals seeking legal assistance. The integration of Rasa enables the AI-Legal Assistant to continuously improve its performance through learning from user interactions, ensuring a dynamic and adaptive tool for the evolving landscape of legal inquiries.

**Work plan**

The development of an AI-Legal Assistant using Rasa chatbot involves a systematic work plan that encompasses various stages to ensure its successful implementation. The initial phase would entail defining the scope and objectives of the legal assistant, identifying key legal domains, and specifying the desired functionalities. Subsequently, data collection and preprocessing are crucial steps, involving the compilation of relevant legal documents, case law, and statutes to train the chatbot effectively. The next stage involves designing and implementing the conversational flow, leveraging the Rasa framework to create intuitive and context-aware interactions. Natural Language Processing (NLP) models need to be integrated to enhance the chatbot's ability to understand and generate legal language. Rigorous testing and validation are imperative to ensure accuracy, reliability, and compliance with legal standards. Continuous refinement and updates should be incorporated based on user feedback and evolving legal landscapes. Furthermore, considerations for data privacy, confidentiality, and ethical use of AI in the legal domain must be integrated into the development process. Finally, the deployment and maintenance plan should be established to guarantee seamless integration into legal workflows, with ongoing support and updates to adapt to changing legal requirements and user needs.

**Methodology**

The development of an AI-Legal Assistant using a Rasa chatbot involves a systematic methodology to ensure the effective integration of natural language processing (NLP) capabilities within a legal context. Rasa, an open-source conversational AI platform, serves as the foundation for creating a chatbot that can understand and respond to legal queries. The methodology begins with defining the scope and objectives of the AI-Legal Assistant, identifying the specific legal domains it will cover, and outlining the desired functionalities. Data collection and preprocessing play a crucial role, involving the acquisition and organization of legal documents, statutes, and case law to train the chatbot. The implementation phase includes designing and training the NLP models within Rasa to comprehend complex legal language and generate contextually relevant responses. Continuous testing and refinement are essential to enhance the chatbot's accuracy and responsiveness. Additionally, integrating legal databases and updates ensures the AI-Legal Assistant remains current and reliable. Finally, user feedback and ongoing maintenance are incorporated to address evolving legal requirements and user needs, contributing to the overall effectiveness and utility of the AI-Legal Assistant powered by the Rasa chatbot.

# **Proposed Solution:**

The proposed solution involves the development of an AI-Legal Assistant utilizing a Rasa chatbot framework. Rasa is an open-source conversational AI platform that enables the creation of intelligent chatbots capable of understanding and responding to user inputs in a natural language. In the context of a legal assistant, the Rasa chatbot would be trained to comprehend legal terminology, processes, and queries. The chatbot could assist users in obtaining information about legal procedures, offer guidance on legal documentation, and provide general legal information. Leveraging machine learning capabilities, the AI-Legal Assistant can continuously improve its performance by learning from user interactions and refining its responses over time. This solution aims to enhance accessibility to legal information, streamline communication between users and legal resources, and contribute to the efficiency of legal support services through the integration of conversational AI technology.

**Chapter 4**

**HARDWARE & SOFTWARE REQUIREMENT**

3.2 Hardware Requirements

* Linux: GNOME or KDE desktop GNU C Library (glibc) 2.15 or later, 2 GB RAM minimum,
* 4 GB RAM recommended, 1280 x 800 minimum screen resolution.
* Windows: Microsoft R Windows R 8/7/Vista (32 or 64-bit) 2 GB RAM minimum, 4 GB RAM
* recommended, 1280 x 800 minimum screen resolution, Intel R processor with support for Intel R
* VT-x, Intel R EM64T (Intel R 64) Execute Disable (XD) Bit functionality

**Software Specification:**

* Windows Operating System.
* MySQL
* Python
* Flask
* Anaconda ,Jupyter, Spyder

**Technologies Used:-**

1. **MySQL:**

Mysql is prestigious as worlds most by and large utilized ascii archive data back-end its most guarantee data for php as php-mysql is most habitually utilized ascii record prearranging data attempt the ui that wamp lamp and xampp workers offer for mysql is ideal and diminishes our work to an outsized degree

1. **Python:**

Python could likewise be a taken item organized basic level language with dynamic derivation its straightforward level in-created information structures got together with unique organization and dynamic restricting sort it outrageously interesting for speedy application advancement what’s more on be utilized as a pre piece or glue language to relate existing components on pythons clear direct to be told accentuation highlights quality by then decreases the cost of program fixes python maintains modules and packs that moves program quality and code utilize the python go-between and what’s more the escalated standard library are offered in give or combined sort to nothing of charge for each and every fundamental stage and wish to be uninhibitedly spread of programmers fall stricken with python because of the misrepresented strength it gives since there is no aggregation step the special stepped area test-investigate cycle is unfathomably expedient work python programs is basic a bug or unfortunate information won’t ever cause a division deformity taking everything into account once the interpreter discovers a blunder it raises an extraordinary case once the program doesn’t get the exception the go-between prints a stack follow a stock level program licenses assessment of local and world elements examination of self-emphatic enunciations setting breakpoints wandering through the code a line at a rapidly on the program is written in python itself vouching for pythons smart power barring generally the quick in view of right a program is to incorporate a few print clarifications to the accessibility the quick modify test-explore cycle makes this simple philosophy dreadfully amazing.

**4)Flask:**

A Flask is a Web Application Framework that is built with Flexibility and Speed In the Mind.Flask is Built in Python , which many data Scientists are familiar with . Flask takes care of the Environment and Project setup involved in web Applications Allowing the Developer to focus on their application rather than thinking about HTTP , routing , dataset etc. Flask allows Data Scientists to create simple Single page Applications and one should Help or look into if they want to create Products for Consumers Flask is a micro web framework written in Python. It is classified as a microframework because it doesn't require particular tools or libraries. There is no database abstraction layer, form validation, or the other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions which will add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and a number of other common framework related tools

Flask was created by Armin Ronacher of Pocoo, a world group of Python enthusiasts formed in 2004.According to Ronacher, the thought was originally an April Fools joke that was popular enough to form into significant application. When Ronacher and Georg Brandl created a bulletin board system written in Python, the Pocoo projects Werkzeug and Jinja were developed. Flask has become popular among Python enthusiasts. As of October 2020, its second most stars on GitHub among Python web-development frameworks, only slightly behind Django, and was voted the foremost popular web framework within the Python Developers Survey 2018.

These are some Important features of the Flask:

1. it is a Development Server

2. Debugger

3.RESTful request dispatching

4. Unicode Based

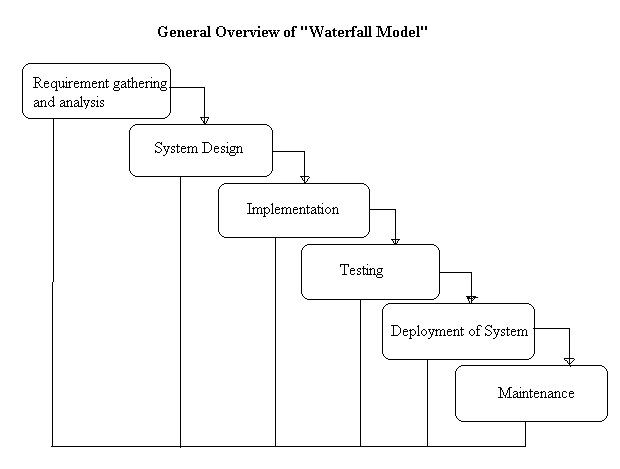
5. Flask have google app engine Compatibility

**Chapter 5**

**PLANNING AND ESTIMATION**

**Software development Life Cycle**

The entire project spanned for a duration of 6 months. In order to effectively design and develop a cost-effective model, the Waterfall model was practiced.

****

**Requirement gathering and Analysis phase:**

this phase started at the beginning of our project. We formed groups and modularized the project. Important points of consideration were

1. Define and visualize all the objectives clearly.

2.Gather requirements and evaluate them

Consider the technical requirements needed and then collect technical specifications of various peripheral components (Hardware) required.

3. Analyze the coding languages needed for the project.

4. Define coding strategies.

5. Analyze future risks / problems.

6. Define strategies to avoid these risks and define alternate solutions to these risks.

7. Check financial feasibility.

8. Define Gantt charts and assign a time span for each phase.

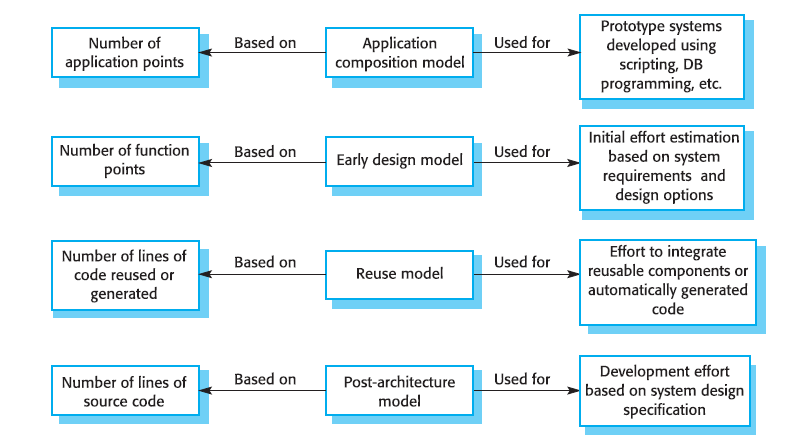
By studying the project extensively we developed a Gantt chart to track and schedule the project. Below is the Gantt chart of our project.

**TimeLineChart**

**Please make changes as per your requirement**

| Task Name | ID | Start Date | Finish Date | Duration | 30/07/2015 **To** 19/08/2015 | 19/08/ **To** 26/08/15 | 27/08/2015 **To** 23/09/2015 | 24/08/2015 To 07/10/2015 | 08/10**To** 15/10 | 08/10**To** 15/10 | 08/10**To** 15/10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement Gathering | 1 | 29/07/15 | 19/08/15 | 3 Weeks |  |  |  |  |  |  |  |
| Problem Definition | 2 | 12/08/15 | 26/08/15 | 1 Week |  |  |  |  |  |  |  |
| Literature Survey | 3 | 19/08/15 | 02/09/15 | 4 Weeks |  |  |  |  |  |  |  |
| Analysis | 4 | 02/09/15 | 02/09/15 | 2 Week |  |  |  |  |  |  |  |
| Flow Chart | 5 | 16/09/15 | 02/09/15 | 1 Week |  |  |  |  |  |  |  |
| Block Diagram | 6 | 30/09/15 | 07/10/15 | 2 weeks |  |  |  |  |  |  |  |
| H/W Specification | 7 | 07/10/15 | 07/10/15 | 1 week |  |  |  |  |  |  |  |
| S/W Specification | 8 | 07/10/15 | 07/10/15 | 1 week |  |  |  |  |  |  |  |

**Cost Estimation**



Cost estimation is done using cocomo model

| cost Drivers | **Ratings** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Very Low | Low | Nominal | High | Very High | Extra High |
| **Product attributes** |  |  |  |  |  |  |
| Required software reliability | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 |  |
| Size of application database |  | 0.94 | 1.00 | 1.08 | 1.16 |  |
| Complexity of the product | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| **Hardware attributes** |  |  |  |  |  |  |
| Run-time performance constraints |  |  | 1.00 | 1.11 | 1.30 | 1.66 |
| Memory constraints |  |  | 1.00 | 1.06 | 1.21 | 1.56 |
| Volatility of the virtual machine environment |  | 0.87 | 1.00 | 1.15 | 1.30 |  |
| Required turnabout time |  | 0.87 | 1.00 | 1.07 | 1.15 |  |
| **Personal attributes** |  |  |  |  |  |  |
| Analyst capability | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 |  |
| Applications experience | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 |  |
| Software engineer capability | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 |  |
| Virtual machine experience | 1.21 | 1.10 | 1.00 | 0.90 |  |  |
| Programming language experience | 1.14 | 1.07 | 1.00 | 0.95 |  |  |
| **Project attributes** |  |  |  |  |  |  |
| Use of software tools | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 |  |
| Application of software engineering methods | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 |  |
| Required development schedule | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 |  |

The Intermediate Cocomo formula now takes the form:

**E=*ai*(KLoC)*(bi)*.EAF**

Using above calculation we found that the total time period of the project is around 6 months, the per month cost comes out to be Rs.12, 000/- so the total comes to be Rs.72, 000/-

**FEASIBILITY STUDY**

This system is possible for all health care departments like science lab hospital and clinic etc and this method can use while not specialists in this field anyone can use who have data concerning using online services which is able to facilitate to use this method any generation folks can use this method in laptop

**TECHNICAL FEASIBILITY**

The framework ought to be assessed from the specialized reason for read first the evaluation of this practicability ought to be upheld a rundown kind of the framework interest inside the provisions of info yield projects and techniques having known an outline framework the examination ought to keep up to suggest the kind of pack required approach building up the framework of running the framework whenever it has been planned. 21 1. Is the existing technology sufficient for the suggested one? 2. Can the system expand if developed? the undertaking should be created indicated the predetermined capacities and execution are accomplished among the limitations the task is created among most recent innovation through the innovation may become old once some measure of some time due to the specific undeniable truth that never form of same code upholds more seasoned variants the framework should in any case be utilized hence there are marginal imperatives included this task the framework has been created exploitation python the undertaking is in fact feasible for advancement

**ECONOMIC FEASIBILITY**

The creating framework ought to be even by worth and benefit. Measures to confirm that exertion is focused on a project, which may give best, come at the most punctual. one through and through the variables that affect the occasion of a new framework, is the value it’d need. The following are an assortment of the necessary cash questions asked all through the starter examination:

1. They conduct a full system investigation.
2. The cost of the hardware and software.
3. The benefits in the form of reduced costs or fewer costly errors.

Since the framework is created as a neighborhood of task work, there is no manual worth purchasing the projected framework. Furthermore every one of the assets are as of now available, it offers an image of the framework is financially feasible for improvement.

**BEHAVIORAL FEASIBILITY**

This incorporates the following inquiries:

1. Is there agreeable help for the clients?
2. Will the arranged framework hurt?

The venture would be useful as an aftereffect of fulfilling the goals once created and introduced. All social perspectives are considered cautiously and presume that the undertaking is typically conceivable

**RISK ANALYSIS PROCESS**

The risk analysis process involves systematically identifying, assessing, and mitigating potential risks that could impact a project, organization, or any endeavor. It typically begins with the identification of potential risks, which can range from financial uncertainties to technological failures or even natural disasters. Once identified, these risks are analyzed to determine their likelihood of occurrence and the potential impact they could have. This analysis helps prioritize risks based on their severity and likelihood, allowing resources to be allocated effectively. Finally, risk mitigation strategies are developed and implemented to reduce the likelihood of negative outcomes or to minimize their impact if they occur. Overall, the risk analysis process is essential for informed decision-making and proactive management of uncertainties in any context.Potential openings could even be delegated regular, specialized, or human dangers. Models include:

**Characteristic Threats:** inner flooding, outer flooding, interior hearth, outside chimney, seismic movement, high breezes, snow and ice storms, emission, cyclone, typhoon, pandemic, torrent , hurricane.

**Specialized Threats:** power disappointment/variance, warming, ventilation or air con disappointment, glitch or disappointment of hardware , disappointment of framework code, disappointment of use code, broadcast communications disappointment, gas spills, interchanges disappointment, atomic aftermath.

**Human** **Threats**:robbery, bomb dangers, theft, blackmail, thievery, defacing, psychological warfare, common problem, synthetic spill, damage, blast, war, natural pollution, radiation tainting, perilous waste, vehicle crash, airdrome nearness, strike (Internal/External), PC wrongdoing. All areas and offices should be encased inside the peril investigation maybe than attempting to sort out real prospects of every fiasco an overall relative game plan of high medium and low is utilized at first to distinguish the probability of the danger happening the possibility investigation also need to affirm the effect of such a likely danger on various capacities or offices inside the association a risk analysis type discovered here pdf format will work with the strategy the capacities or divisions can shift by kind of association the arranging strategy ought to set up and live the possibility of every single expected danger and in this way the effect on the association if that danger happened to attempt to this each division should be investigated severally in spite of the fact that the chief framework is furthermore the one most serious danger it isn’t the solitary vital concern indeed even inside the first programmed associations a few offices will not be handled or programmed inside the smallest degree in totally programmed divisions essential records stay outside the framework as lawful records pc information programming bundle hang on diskettes or supporting documentation for data section the effect is evaluated as 0 no effect or break in tasks 1 noticeable effect break in activities for as long as eight hours 2 mischief to instrumentation and additionally offices break in tasks for eight 48 hours 3 major damage to the instrumentation or potentially offices break in tasks for very 48 hours all base camp or potentially pc focus capacities ought to be resettled bound suspicions is also important to consistently apply evaluations to every possible danger

**Functional requirements :**

1. System should have sufficient internet to fetch the data from the server.
2. The system will acquire all data on a daily basis.
3. System should be able to match required configurations.
4. Database should be updated with the latest values.
5. The system should have to display

**Non-functional requirements :**

1.A legal assistant AI system must meet several non-functional requirements to ensure its effectiveness and usability in assisting legal professionals.

2.These requirements include reliability, ensuring that the AI consistently provides accurate and relevant information without errors, as well as scalability to accommodate varying workloads and data volumes. Additionally, the system should prioritize security, safeguarding sensitive legal information and maintaining compliance with privacy regulations.

3.Usability and user experience are paramount, necessitating an intuitive interface and seamless integration into existing workflows. Furthermore, the AI should exhibit responsiveness, delivering timely assistance and minimizing latency in its interactions. Finally, the system must possess robustness, capable of handling diverse legal tasks and adapting to evolving requirements and scenarios within the legal domain.

**Chapter 6**

**TESTING**

Testing:

An AI legal assistant is a software tool designed to support legal professionals by automating tasks, conducting legal research, and providing analysis of case law and statutes. These assistants utilize artificial intelligence algorithms to process large volumes of legal data efficiently and accurately, helping lawyers streamline their workflow and make more informed decisions. By leveraging natural language processing and machine learning capabilities, AI legal assistants can assist with document review, contract analysis, due diligence, and other aspects of legal work, ultimately saving time and reducing the potential for errors.

Error and exception handling:

Sure! Testing error and exception handling in a Rasa chatbot involves thoroughly examining how the bot responds to unexpected inputs or situations. This process includes intentionally feeding the bot incorrect or ambiguous messages to see how it handles them. By doing so, developers can identify and address potential issues such as crashes, incorrect responses, or failure to understand user inputs. Effective error and exception handling ensures that the chatbot can gracefully recover from unexpected scenarios, providing a smoother and more reliable user experience.

FeatureExtraction:  
 Sure, here's a brief summary: The AI legal assistant, RasaChatbot, is equipped with a feature extraction capability. This means it can analyze and extract relevant features or information from legal documents, conversations, or other inputs. This feature enables the chatbot to better understand and respond to legal queries, making it a valuable tool for legal professionals and individuals seeking legal assistance.

**Chapter 7**

**Design & Implementation**

**SYSTEM IMPLEMENTATION**

**ER -DIAGRAM:**

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them.

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER modeling is something regarded as a complete approach to design a logical database schema. This is incorrect because the ER diagram is just an approximate description of data, constructed through a very subjective evaluation of the information collected during requirements analysis.

ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers.

* **Entity**

An entity is an object or component of data. An entity is represented as a rectangle in an ER diagram.  
For example: Student and College and these two entities have many to one relationship as many student studies in a single college.

An entity that cannot be uniquely identified by its own attributes and relies on the relationship with another entity is called a weak **entity**. The weak entity is represented by a double rectangle.

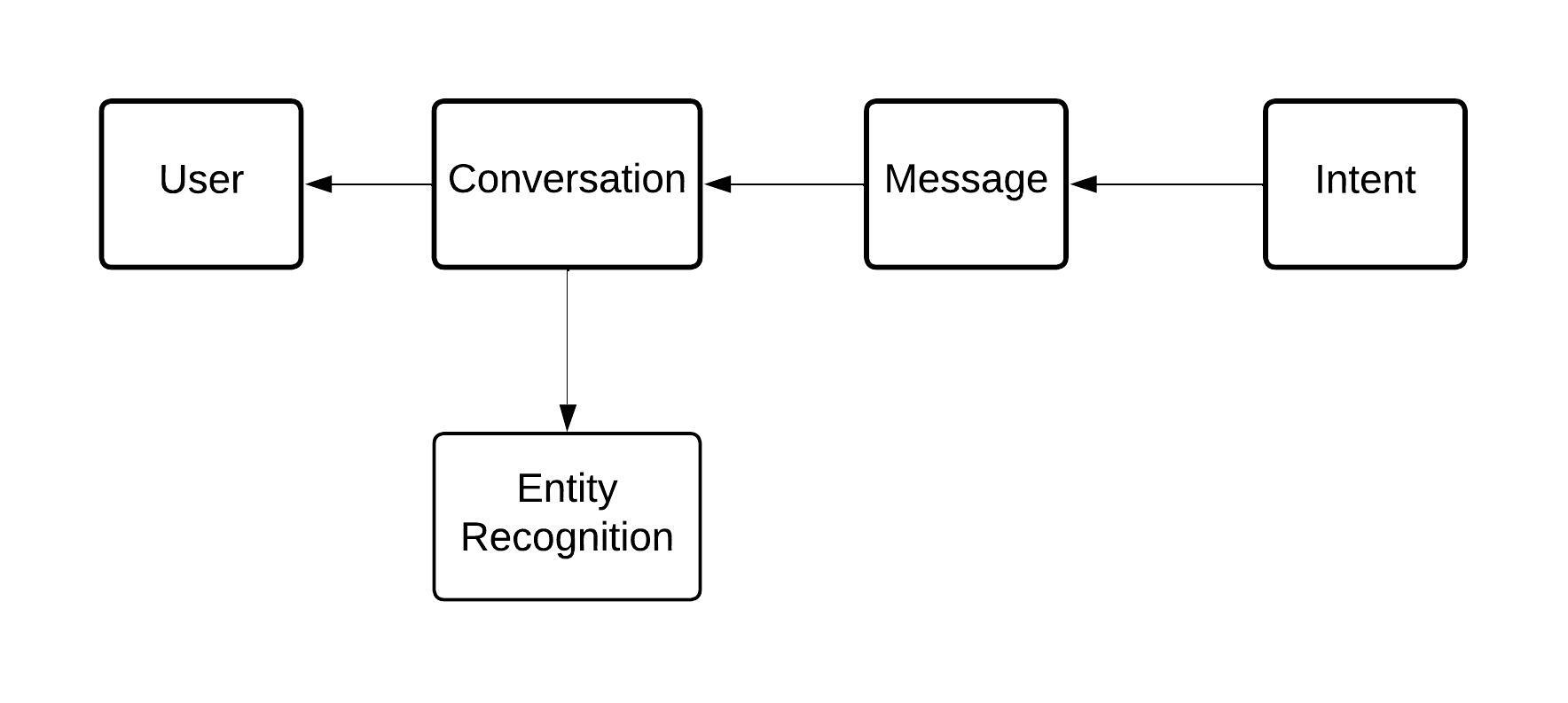
* **Attribute**

An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

1. Key attribute  
2. Composite attribute  
3. Multivalued attribute  
4. Derived attribute

* **Relationship**

A relationship is represented by diamond shape in the ER diagram, it shows the relationship among entities. There are four types of relationships:  
 1. One to One  
 2. One to Many  
 3. Many to One  
 4. Many to Many

****

**FLOWCHART:**

Flow diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, Flow diagrams can be used to describe the business and operational step-by- step workflows of components in a system. An activity diagram shows the overall flow of control. A Flow diagram shows the overall flow of control. Flow diagrams are constructed from a limited repertoire of shapes, connected with arrows.

Flow diagrams are constructed from a limited repertoire of shapes, connected with arrows.

The most important shape types:

● The rectangle represents Flow .

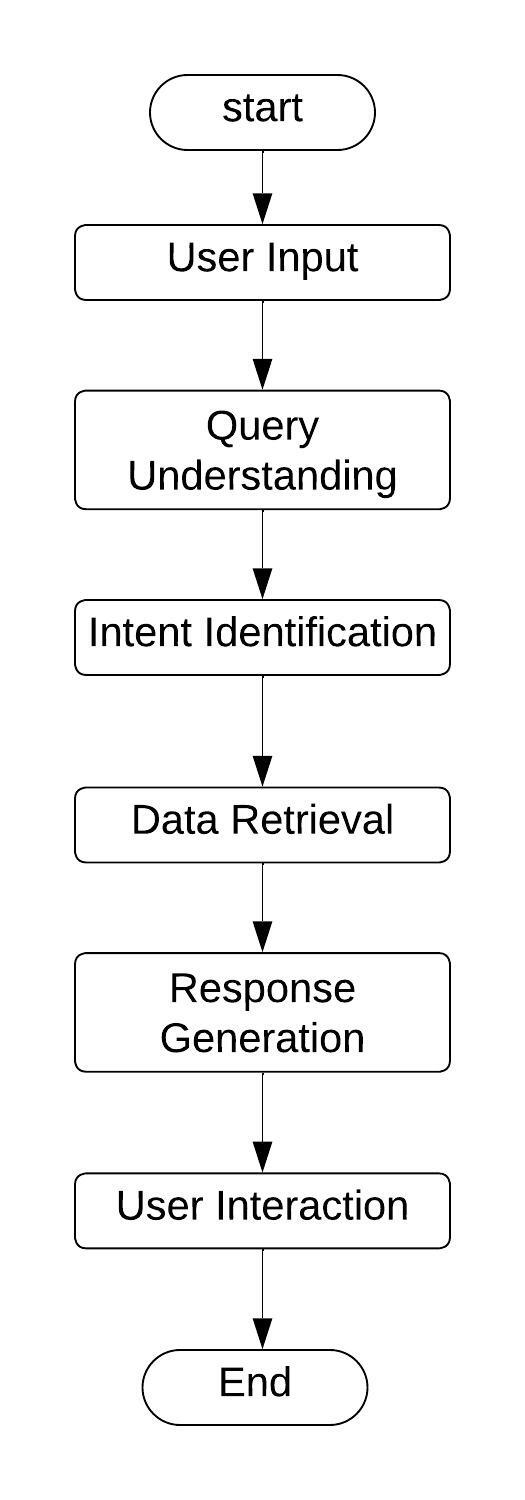
● Diamonds represent decisions.

● Bars represent the start (split) or end (join) of concurrent activities.

● A rectangle represents the start (initial state) of the workflow.

● An end rectangle represents the end (final state).

● Arrows run from the start towards the end and represent the order in which activities happen.

****

**Context level Data Flow Diagram:**

Data Flow Diagram (DFD) is a graphical representation of data flow in any system. It is capable of illustrating incoming data flow, outgoing data flow and store data. There is a major difference between data flow diagrams and flowchart.. Data flow diagrams illustrate flow of data in the system at various levels. Data flow diagram does not have any control or branch elements.Data flow diagram describes anything about how data flows through the system.Sometimes people get confused between data flow diagram and flowchart. The flowchart illustrates flow control in program modules

**Components of Data Flow Diagram**:

**Entities:**

Entities include source and destination of the data. Entities are represented by a rectangle with their corresponding names.

**Process:**

The tasks performed on the data are known as processes. Process is represented by a circle. Somewhere round edge rectangles are also used to represent the process.

**Data Storage:**

Data storage includes the database of the system. It is represented by a rectangle with both smaller sides missing or in other words within two parallel lines.

**Data Flow:**

The movement of data in the system is known as data flow. It is represented with the help of an arrow. The tail of the arrow is the source and the head of the arrow is the destination.

DFD Level 0:-















**Activity diagram:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by- step workflows of components in a system. An activity diagram shows the overall flow of control. An activity diagram shows the overall flow of control. Activity diagrams are constructed from a limited repertoire of shapes, connected with arrows.

Activity diagrams are constructed from a limited repertoire of shapes, connected with arrows.

The most important shape types:

● Rounded rectangle represents activities.

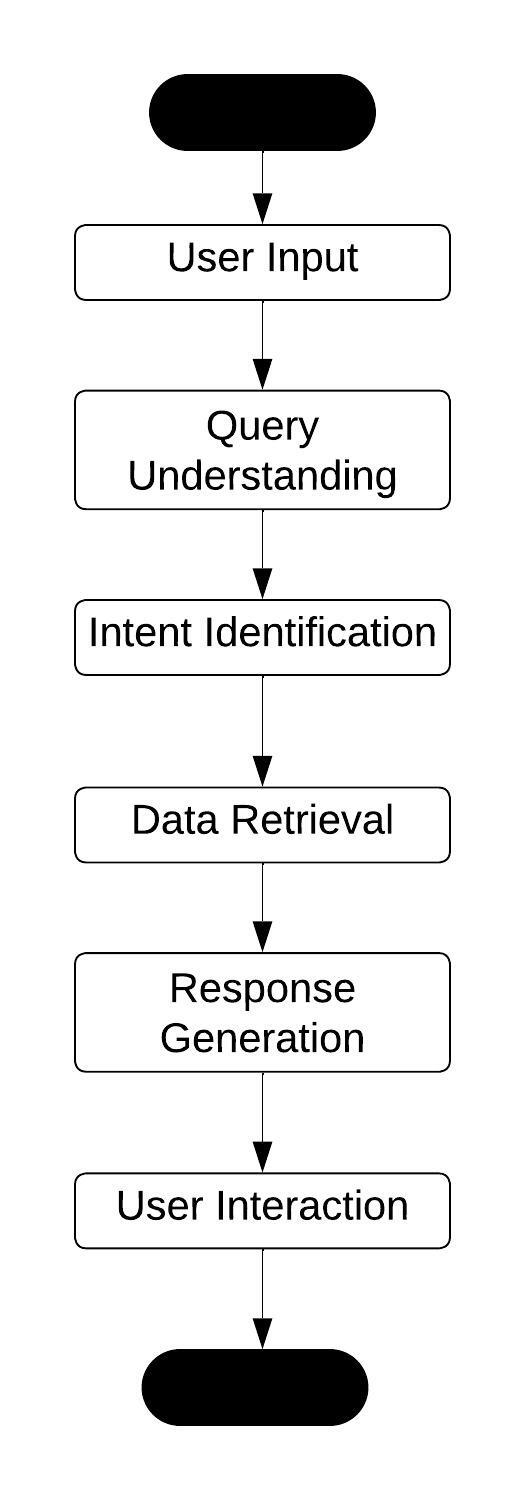
● Diamonds represent decisions.

● Bars represent the start (split) or end (join) of concurrent activities.

● A black circle represents the start (initial state) of the workflow.

● An encircled black circle represents the end (final state).

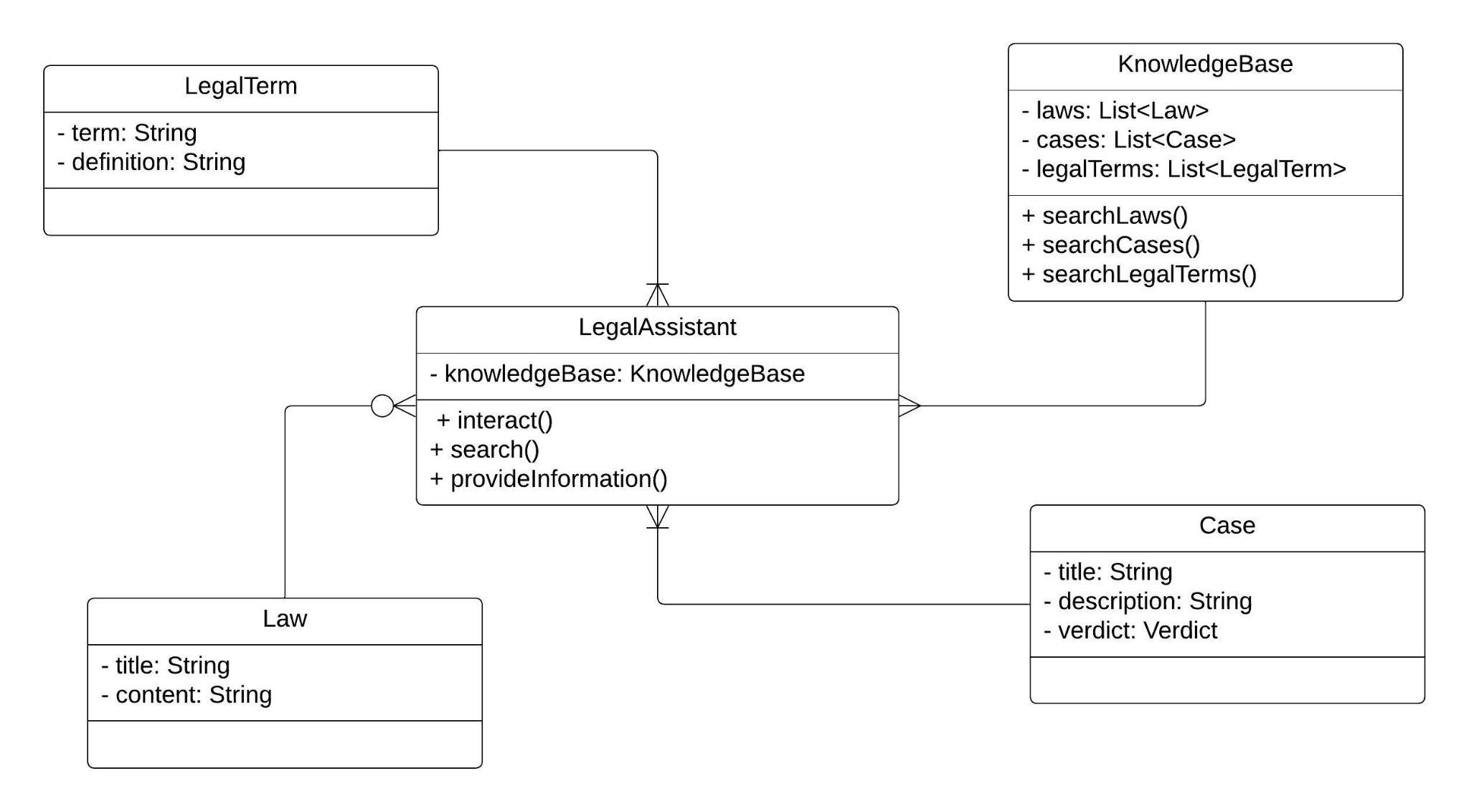
● Arrows run from the start towards the end and represent the order in which activities happen.



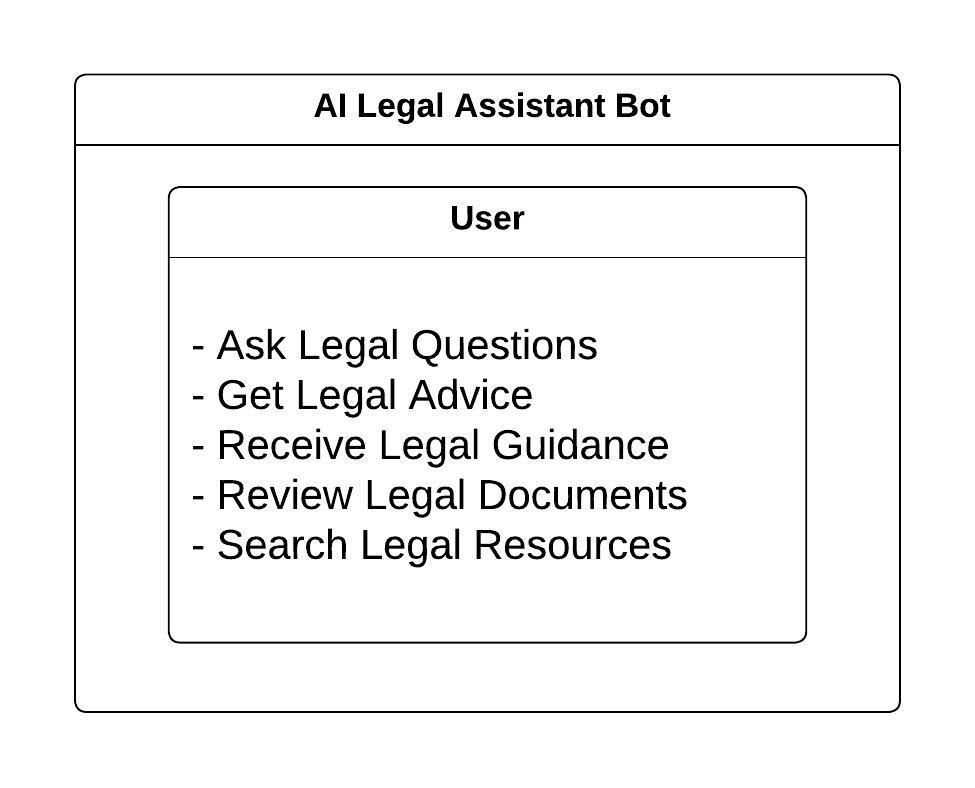
Class diagram:

It is a model which is used to show the classes constituting a system and their interrelationship. It is based on UML. Only the important attributes and methods are shown in Class diagrams. In the initial period of analysis, the important attributes of the classes, which must be captured and the functionalities provided by the class may not be very clear. As the analysis progresses, the attributes and methods may be added. If more focus is on interrelationships of classes, then the attributes and methods may not be shown in the class diagram.

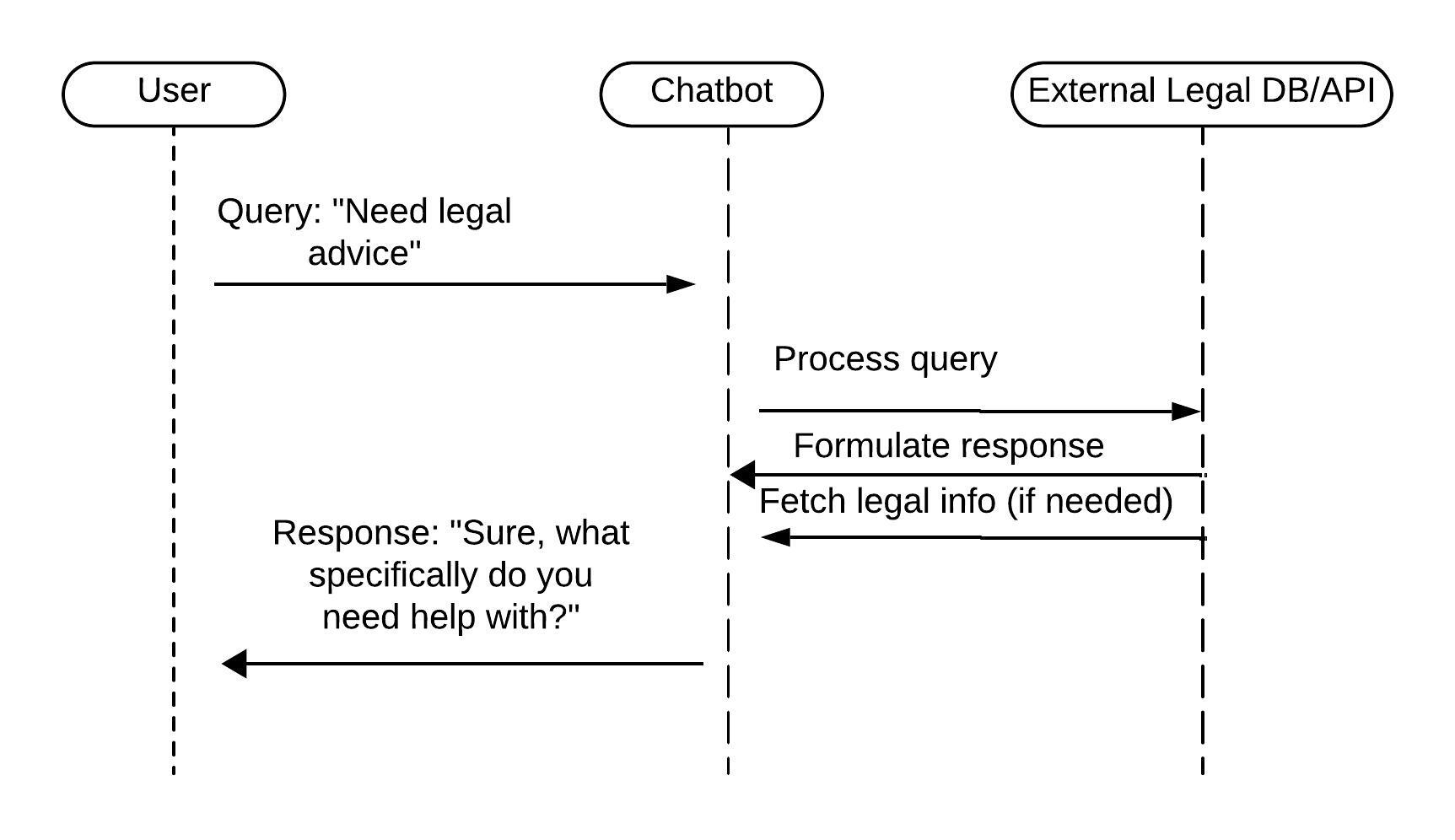
The class diagram is used to identify and classify the objects which constitute a system. It also includes the important attributes of the objects which must be captured.



**Use Case Diagram:**

****

**Sequence Diagram:**

****

**Chapter 8**

**ADVANTAGES**

An AI legal assistant like Rasachatbot offers several advantages in the legal domain. Firstly, it enhances efficiency by streamlining repetitive tasks such as document review, legal research, and drafting standard contracts. This allows legal professionals to focus on higher-value activities, thereby increasing productivity and reducing costs. Secondly, it improves accessibility to legal information by providing instant responses to queries, enabling individuals to gain quick insights into legal matters without the need for extensive research or consultation. Additionally, AI legal assistants can enhance accuracy and consistency in legal work, minimizing the risk of errors or oversights. Moreover, they facilitate collaboration among legal teams by centralizing information and enabling seamless communication and document sharing. Overall, the adoption of AI legal assistants like Rasachatbot stands to revolutionize the legal industry by optimizing workflows, improving accessibility, and elevating the quality of legal services provided.

**Chapter 10**

**Results and Discussion**

**& CONCLUSION**

**Results and Discussion:**

An AI legal assistant, RasaChatbot, has been developed to facilitate legal research and provide guidance in navigating legal complexities. Through its natural language processing capabilities, RasaChatbot offers users a streamlined approach to accessing legal information and assistance. Results indicate that this AI solution has the potential to significantly enhance efficiency in legal research and decision-making processes. Moreover, discussions surrounding its implementation underscore the importance of ensuring accuracy, privacy, and ethical considerations in the development and deployment of such technologies within the legal domain. Overall, RasaChatbot represents a promising advancement in leveraging AI to support legal professionals and individuals in their legal inquiries and endeavors.

**Feature**

The AI Legal Assistant, RasaChatbot, offers a comprehensive set of features designed to streamline and enhance legal processes. Its primary functions include providing legal advice, assisting with document drafting and review, managing case files, scheduling appointments, and facilitating communication between legal professionals and clients. With its advanced natural language processing capabilities, RasaChatbot can efficiently analyze complex legal queries and provide accurate responses, significantly reducing the time and effort required for research and analysis. Its user-friendly interface and customizable options make it a valuable tool for legal professionals seeking to improve efficiency and productivity in their practice.

**CONCLUSION:**

The AI legal assistant, RasaChatBot, offers a comprehensive solution for legal inquiries, streamlining processes and enhancing efficiency within the legal domain. Through its advanced natural language processing capabilities, RasaChatBot simplifies complex legal jargon and assists users in navigating legal complexities with ease. By providing accurate and timely responses to legal queries, it facilitates informed decision-making and ensures compliance with relevant laws and regulations. RasaChatBot stands as a testament to the transformative potential of AI in the legal field, offering invaluable support to legal professionals and individuals alike.

**Chapter 11**

**BIBLIOGRAPHY**

**Bibliography:**

[1] W. W. Widiyanto, U. Rizki, and E. Susena, “Increased accuracy

of sequence-to-sequence models with the CNN algorithm for

multi response ranking on travel service conversations based on

chat history,” Jurnal INFOTEL, vol. 12, no. 2, pp. 39–44, May

2020.

[2] E. Adamopoulou and L. Moussiades, “Chatbots: History,

technology, and applications,” Machine Learning with

Applications, vol. 2, p. 100006, Dec. 2020.

[3] S. M. Suhaili, N. Salim, and M. N. Jambli, “Service chatbots: A

systematic review,” Expert Systems with Applications, vol. 184.

Elsevier Ltd, Dec. 2021.

[4] G. Daniel and J. Cabot, “The software challenges of building

smart chatbots,” In Proc. IEEE International Conference on

Software Engineering, May 2021, pp. 324–325.

[5] F. Brachten, T. Kissmer, and S. Stieglitz, “The acceptance of

chatbots in an enterprise context – A survey study,” International

Journal of Information Management, vol. 60, no. 102375, Oct.

2021.

[6] J. Doshi, “Chatbot user interface for customer relationship

management using NLP models,” In Proc. IEEE International

Conference on Artificial Intelligence and Machine Vision, Jan.

2022, pp. 1–4.

[7] S. Panda and R. Chakravarty, “Adapting intelligent information

services in libraries: a case of smart AI chatbots,” Library Hi Tech

News, vol. 39, no. 1, pp. 12–15, Feb. 2022.

[8] A. Janssen, D. Rodríguez Cardona, J. Passlick, and M. H.

Breitner, “How to Make chatbots productive – A user-oriented

implementation framework,” International Journal of Human

Computer Studies, vol. 168, Dec. 2022.

[9] C. V. Misischia, F. Poecze, and C. Strauss, “Chatbots in customer

service: Their relevance and impact on service quality,” in

Procedia Computer Science, Elsevier B.V., 2022, pp. 421–428.

[10] I. K. F. Haugeland, A. Følstad, C. Taylor, and C. A. Bjørkli,

“Understanding the user experience of customer service

chatbots: An experimental study of chatbot interaction design,”

International Journal of Human Computer Studies, vol. 161, no.

102788, May 2022.

[11] C. Berti and A. L. Vecchia, “Temporal trend of child stunting

prevalence and food and nutritional surveillance system,” Journal

de Pediatria, vol. 99, no. 2, pp. 99–100, Apr 2023.

[12] F. Oktaviana, “Early detection of the risk of stunting in pregnant

women and its recommendations,” In Proc. IEEE International

Symposium on Community-centric Systems, Oct. 2020, pp. 1–6.

[13] J. Fanzo, et al. “2018 Global nutrition report: Shining a light

to spur action on nutrition,” School of Health & Psychological

Sciences, London, UK: Global Nutrition Report, 2018.

[14] S. N. M. S. Pi, and M. A. Majid, “Components of smart chatbot

academic model for a university website,” In Proc. IEEE Emerging

Technology in Computing, Communication and Electronics, Feb.

2021, pp. 1–6.

[15] T. Matsuo and S. Iwamitsu, “Sustainable city planning and public

administration assisted by green AI: attendant legal challenges

under Japanese law,”

**Chapter 11**

**SCREENSHOTS**

**Chapter 12**

**SOURCE CODE**