**Batch:**-B-2 **Roll No:**-16010122151

**Exp No:- 2**

**Title:** Identification of data and hardware software Requirement (Analysis phase).

**Objective:** To understand & identify hardware and software, usage of Tools needed to meet the desired specification of the project.

**Expected Outcome of Experiment:**

| Course Outcome | After successful completion of the course students should be able to |
|---|---|
| CO 2 | Identify the various hardware and software, usage of Tools needed to meet the desired specification. |

**Books/ Journals/ Websites referred:**

**1.** Wikipedia (https://www.wikipedia.org)
**2.** GitHub (https://github.com)
**3.** Stanford Encyclopedia of Philosophy (https://plato.stanford.edu)

**Describe the need of this stage in project:**

Project Requirements specify what features a product should include and how those features should work. They help to define the test criteria, which is vital for verification and validation.

At the beginning of the software development process, collating and analysing the requirements is one of the first tasks to be done. This usually takes place between the client, software engineer, and tester. All three stakeholders, must come to an agreement as to what the final product should do and how. In addition, to understanding how the product will be verified and validated according to the requirements set out.

**Types of requirements:**

There are several types of requirements that should be considered when it comes to designing or modifying software. They each depend on what the product is. These are: business requirements, software requirements, and hardware requirements.

Business requirements:
These outline the business case and reason for making the product in the first place. If a company doesn't have a business case that stacks up then why is the company investing money and time into taking the project on further.

Also known as stakeholder requirements, they describe the characteristics of the proposed product from the point of view of the end user. They don't define what the system should do or how it should be done, but instead they define the 'why' requirements. For example, why does this customer need your product? It sets the tone and vision for the product.

**Software requirements:**
These can be categorised into **Non-Functional and Functional requirements**. The difference is as follows:

Functional requirements are the 'what' requirements. They outline what is the system designed to do. For example, say you were a medical company looking to develop a COVID testing device. The types of functional requirements, which might exist could be that the product must be able to identify a COVID test or the product must display a positive or negative test result to the user.

Non-Functional requirements are the 'how' requirements. They outline how the system will do what it is designed to do. For example, the product will run a COVID test within 20 minutes and the device will send results via Bluetooth to the end user's mobile phone. Both set of requirements are important as they provide input into what will be required for development and testing.

**Hardware requirements:**
These are the requirements that are required to develop and create a hardware device. Sometimes a product is completely software-based and so they may not be needed. But if your company is making a physical product then there are many aspects to consider before designing and making the device.

There are many aspects such as what processor speed is required, how much memory is needed, what shape will the product take that all need to be reviewed. Hardware and software requirements need to be considered together to determine the software compatibility for the system being designed.

**Answer the following based on your case study:**

1. **Identify data:-**
   - **Legal Knowledge Base:-** This includes a comprehensive collection of legal information, statutes, regulations, case law, and legal principles relevant to the chatbot's domain. This data would need to be curated and structured for easy retrieval by the chatbot.
   - **User Queries and Interactions:-** Data on user queries, interactions, and feedback with the chatbot are crucial for improving its performance over time. This data helps in understanding user needs, refining the chatbot's responses, and identifying areas for improvement.
   - **Legal Document Templates:-** If the chatbot assists in drafting legal documents, it would need access to templates and examples of various legal documents such as contracts, agreements, leases, etc.
   - **Updates and Changes in Laws:-** Data sources that provide updates and changes in laws and regulations relevant to the chatbot's domain are essential for keeping the information provided by the chatbot up-to-date and accurate.
   - **User Profiles**:- Depending on privacy and data security considerations, the chatbot may collect and store user profiles to provide personalized legal advice and recommendations. This data must be handled securely and ethically.
   - **Chatbot Analytics**:- Metrics related to the chatbot's performance, such as response times, user satisfaction ratings, usage patterns, and frequently asked questions, are valuable for evaluating and improving the chatbot's effectiveness.

2. **Identify software and hardware needed.**

   **Software Needed:-**

   **1. Programming Languages:**

   - Python: For building the chatbot logic and backend functionality.
   - JavaScript/HTML/CSS: For developing the frontend interface of the chatbot.

   **2. Chatbot Development Frameworks/Libraries:-**

   **- Natural Language Processing (NLP) Libraries:-**

   - NLTK (Natural Language Toolkit): For text processing and NLP tasks.
   - SpaCy: For advanced NLP tasks such as entity recognition and text analysis.

   **- Chatbot Frameworks:-**

- Rasa: Open-source framework for building conversational AI chatbots.
- Dialogflow: Google's NLP and conversation management platform.

**3. Web Development Tools:-**

- IDE (Integrated Development Environment): Examples include PyCharm, Visual Studio Code for Python, and Sublime Text for web development.
- Version Control: Git for managing code versions and collaboration.
- Web Browser: Chrome, Firefox, etc., for testing the chatbot frontend**.**

**4. Database:-**

- MongoDB, MySQL, PostgreSQL: For storing user data, chatbot logs, and legal information.

**Hardware Needed:-**

**1. Development Machine:-**

- Computer with sufficient processing power and memory to run development tools and frameworks smoothly.
- Operating System: Windows, macOS, or Linux.

**2. Server/Cloud Hosting:-**

- Depending on deployment needs, a server or cloud hosting platform such as AWS, Google Cloud Platform, or Heroku may be required to host the chatbot backend and database.

**3. Internet Connection:-**

- Stable internet connection for accessing online resources, deploying updates, and testing the chatbot.

**4. Optional:-**

- Webcam and microphone: If the chatbot supports voice interactions, these may be needed for testing.
- Mobile Device: For testing mobile responsiveness of the chatbot interface.

**3. Give functional and non-functional requirements.**

**Functional Requirements:-**

**User Interaction:-**

- Allow users to ask legal questions in natural language.
- Provide clear and accurate responses based on the legal knowledge base.
- Guide users through legal processes with step-by-step instructions.
- Assist in creating legal documents by generating templates and explanations.
- Handle follow-up questions and maintain context during conversations.

**Legal Information Retrieval:-**

- Retrieve relevant legal information from the knowledge base based on user queries.
- Support searching and browsing legal documents, statutes, and case law.
- Provide citations and references for legal information presented.

**Personalization:-**

- Offer personalized legal advice based on user profiles and specific situations.
- Remember user preferences, history, and previous interactions for continuity.
- Provide recommendations and suggestions tailored to the user's needs.

**Updates and Notifications:-**

- Notify users about changes in laws and regulations relevant to their interests.
- Provide alerts for upcoming legal deadlines, events, or updates.

**Integration:-**

- Integrate with external legal databases, APIs, and services for enriched information.
- Support integration with legal document management systems for document creation and storage.

**Non-Functional Requirements:-**

**Performance:-**

- **Response time:-** Provide quick and responsive interactions with minimal latency.
- **Scalability:-** Handle a large number of concurrent users and scale resources as needed.
- **Reliability:-** Ensure high availability and minimal downtime for uninterrupted service.

**Security and Privacy:-**

- **Data Security:-** Encrypt sensitive user data and communications to protect confidentiality.
- **Compliance:-** Adhere to legal and regulatory requirements for data privacy and security.
- **Authentication and Authorization-:** Implement secure user authentication and access control mechanisms.

**Usability:-**

- **User Interface:-**Design an intuitive and user-friendly chatbot interface for easy navigation.
- **Accessibility:-** Ensure accessibility for users with disabilities, including screen readers and keyboard navigation.
- **Multilingual Support:-** Provide support for multiple languages to cater to a diverse user base.

**Scalability:-**

- Handle a growing volume of user interactions and data without performance degradation.
- Support multi-channel interactions, including web, mobile, and messaging platforms.

**Maintenance and Support:-**

- **Easy Maintenance**:- Design the chatbot architecture for ease of maintenance and updates.
- **Support:-** Provide comprehensive documentation, tutorials, and user support for troubleshooting and assistance.

**Post Lab Activities (with reference to your tool):**

1. What are the tools used for the Software development life cycle?

**Planning and Requirements Analysis:-**

- **Google Sheets:-** For task tracking and simple project planning.
- **Microsoft Word or Google Docs:-** For documenting requirements and user stories.

**Design and Prototyping:-**

- **Pen and Paper:-** Initial sketches and wireframes.
- **Canva:-** Simple graphic design tool for UI mockups.

**Development:-**

- **Visual Studio Code:-** Lightweight code editor for coding.
- **GitHub:-** Version control and collaboration platform for managing code.

**Testing:-**

- **Manual Testing:-** Testing the application manually without automated tools.
- **Excel/Google Sheets:-** For test case management and tracking.

**Deployment and Operations:-**

- **Manual Deployment:-** Uploading files to a server manually.
- **Built-in hosting platforms:-** Platforms like Netlify for hosting web applications.

**Documentation and Collaboration:-**

- **Microsoft Word/Google Docs:-**For project documentation.
- **Email and Messaging Apps:-** For team communication and collaboration.