**Contents**

[Introduction 3](#_Toc162223422)

[Research Methodology 3](#_Toc162223423)

[Literature Search 4](#_Toc162223424)

[Selection Criteria 4](#_Toc162223425)

[Background Research 4](#_Toc162223426)

[History of NLP 6](#_Toc162223427)

[Fundamentals of NLP 7](#_Toc162223428)

[Syntax vs. Semantics 7](#_Toc162223429)

[Tokenization 8](#_Toc162223430)

[Part-of-Speech Tagging 9](#_Toc162223431)

[Name Entity Recognition 10](#_Toc162223432)

[Dependency Parsing 11](#_Toc162223433)

[Lemmatization and Stemming 11](#_Toc162223434)

[Lemmatization 11](#_Toc162223435)

[Stemming 12](#_Toc162223436)

[Comparative Analysis 12](#_Toc162223437)

[Machine learning in NLP 12](#_Toc162223438)

[The Role of NLP in Enhancing Access to Justice 13](#_Toc162223439)

[Identifying and Mitigating Bias 14](#_Toc162223440)

[Privacy and Security Concerns in Legal NLP Applications 14](#_Toc162223441)

[Current Trends and Future Directions of NLP in law 15](#_Toc162223442)

[Project Proposal 16](#_Toc162223443)

[Project Details: 16](#_Toc162223444)

[Project Specification: 16](#_Toc162223445)

[Project User Stories 17](#_Toc162223446)

[High Level System Design: 18](#_Toc162223447)

[Requirement Analysis: 20](#_Toc162223448)

[MoSCoW Analysis 20](#_Toc162223449)

[Project System Requirement: 20](#_Toc162223450)

[Project Solution 22](#_Toc162223451)

[Assumptions, Constraints, Risks: 22](#_Toc162223452)

[Solution Description: 24](#_Toc162223453)

[Solution Delivery Approach: 24](#_Toc162223454)

[Project Management Approach: 24](#_Toc162223455)

[References 26](#_Toc162223456)

# Introduction

The way people get legal help has changed a lot over time. There was a time when getting legal advice was hard and expensive for most people. As the world changed, so did the need for legal help, which became more complicated. Technology has played a big role in changing how we get legal support today. Now, legal help is a key part of dealing with many problems in life, and it affects how fair and just people think the world is.

Thanks to technology, getting legal advice is much easier than before. Smartphones and apps have gone from just being ways to talk to each other to important tools for handling different parts of our lives, including legal issues. This technology lets people find legal information and help easily, giving them the power to handle their legal matters more actively. But, there's a problem: many good legal apps cost money every month, or people have to use many different apps to find what they need. This can make the whole process frustrating and confusing.

My project, Litigat8, is here to make things better. I've created Litigat8 to be a complete source of legal support, especially for issues between landlords and tenants. Whether it's understanding your rights as a tenant, knowing what a landlord can or cannot do, or getting advice on how to handle a dispute, Litigat8 is designed to help. My app offers easy-to-understand advice and resources for tenant and landlord laws.

Litigat8 is more than just information. It's a tool that helps users track their questions, organize their documents, and manage their legal needs easily. I made this app to be a helper and guide in the complex world of tenant and landlord laws. By making legal support simple, direct, and accessible, my goal is to make everyone feel confident and informed about their rights and responsibilities.

With Litigat8, I aim to change the way legal assistance is provided, making it not just easy to access but also supportive and empowering. I want to create a future where everyone, no matter their situation, can understand and use the law to their advantage. Litigat8 isn't just an app; it's a step towards a world where legal help is a tool for fairness and justice for everyone

Research Methodology

This section delineates the methodical approach employed for conducting a comprehensive literature search, selection, and subsequent analysis focusing on the role of Natural Language Processing (NLP) in improving access to justice.

## Literature Search

The literature search was meticulously performed across various academic databases to ensure a broad and thorough collection of relevant studies. The databases queried included Google Scholar, IEEE Xplore, ScienceDirect, SpringerLink, and the ACM Digital Library. To augment the scope of this research, additional sources were located through the references of seminal articles. The search strategy employed combinations and permutations of key terms such as "Natural Language Processing," "NLP in law," "access to justice," "legal informatics," "automated legal assistance," "bias in NLP applications," "privacy in legal NLP," and "future of NLP in law." Boolean operators (AND, OR) were employed to refine the searches, ensuring a focused retrieval of pertinent literature.

## Selection Criteria

**Inclusion Criteria:**

Peer-reviewed articles and conference papers written in English.

Studies that specifically explore the application of NLP technologies in the legal domain or for enhancing access to justice.

Publications that provide insights into the current trends, challenges, and future directions of NLP in legal applications.

**Exclusion Criteria:**

Non-peer-reviewed articles, grey literature, and opinion pieces.

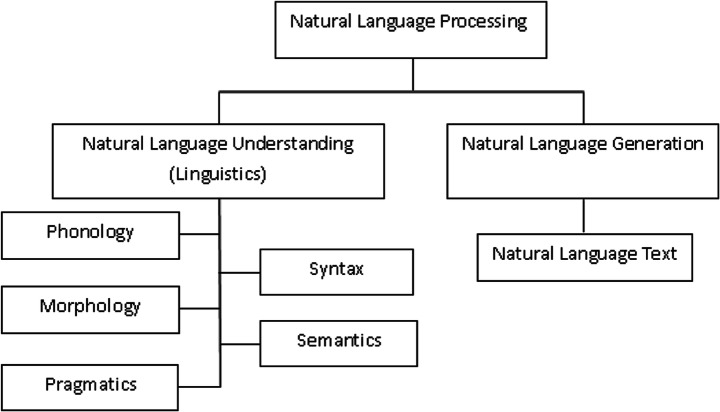
Research that does not directly address NLP's impact on legal processes or access to justice.

Articles without empirical data, clear results on NLP's effectiveness, or a detailed methodology.

An initial screening based on titles and abstracts was conducted to evaluate each article's relevance to the research objectives. This was followed by a full-text review to ensure that the selected studies met the inclusion criteria. Duplicates were removed, and the selected literature was organized for in-depth analysis.

# Background Research

Natural Language Processing (NLP) serves as a critical bridge between human communication and computer understanding, positioning itself as a foundational pillar in both Artificial Intelligence (AI) and linguistics. Designed to streamline interactions between humans and computers, NLP aims to endow computers with the ability to understand human language with ease, bypassing the need for individuals to master complex computer languages. This discipline emerged from the aspiration to converse with computers using our natural vernacular, simplifying technological interactions. NLP divides into two main branches: Natural Language Understanding (NLU) and Natural Language Generation (NLG), focusing on comprehension and the generation of human-like text, respectively. Linguistics, or the scientific study of language, is crucial to NLP, encompassing everything from phonology (the sounds of language) to semantics (the study of meaning).



The groundwork for NLP was laid in the mid-20th century with the development of syntactic theories by Noam Chomsky, revolutionizing syntax in theoretical linguistics and setting the stage for future advancements in both NLP and NLG (Chomsky, N., 1965). Historically, the field has been primarily driven by computer scientists, but it has also drawn attention from linguists, psychologists, and philosophers. NLP enhances our comprehension of human language by merging theories and techniques aimed at facilitating natural language communication with computers. It encompasses a variety of tasks, including automatic summarization, machine translation, and optical character recognition, all of which have direct real-world applications.

The evolution from rule-based to statistical methods and, more recently, to deep learning approaches, like those exemplified by BERT and GPT, has significantly advanced the capabilities of NLP systems. These methodologies demonstrate an enhanced level of understanding and text generation, making NLP a key player in modern technological discussions (Sutskever, I., Vinyals, O., and Le, Q.V., 2014).

Moreover, NLP's role extends to improving human-computer interfaces, automating customer service, delivering personalized content, and performing sentiment analysis. In sectors such as legal services, NLP can streamline document analysis, simplify legal research, and make expert advice more accessible to the general public, showcasing its vast potential to democratize specialized knowledge.

Despite these advancements, NLP faces challenges like language ambiguity, understanding context, and addressing the nuances of cultural language variations. Addressing these issues remains an active area of research, aiming to enhance the sophistication of AI in interpreting human language.

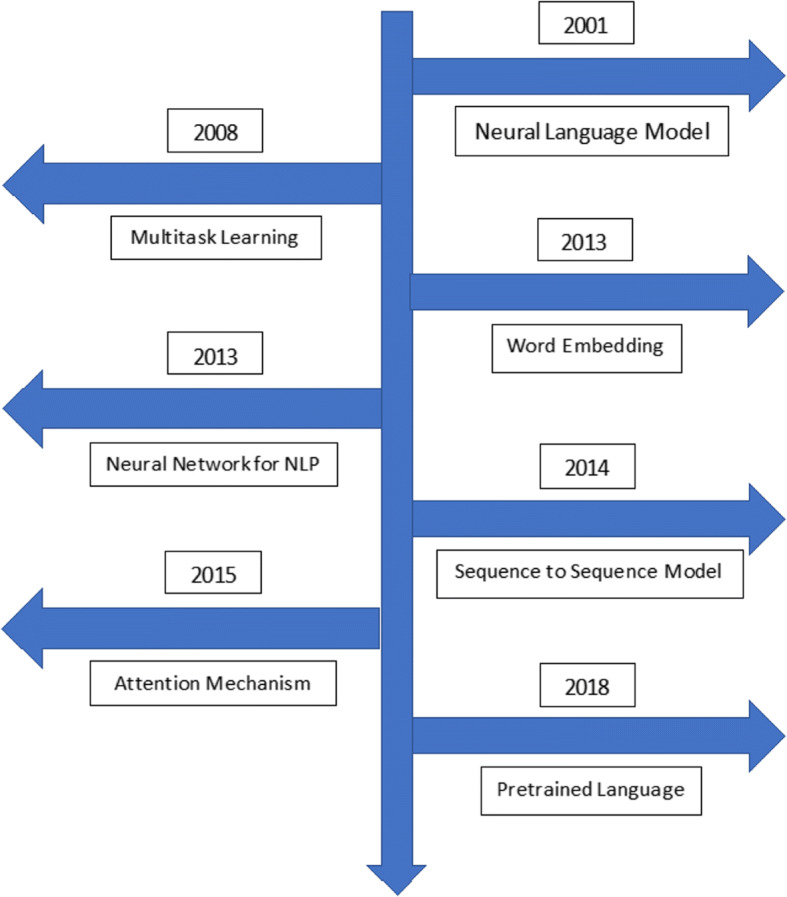
# History of NLP

The history and evolution of Natural Language Processing (NLP) can be traced back to the mid-20th century, marking a journey filled with significant milestones, technological advancements, and interdisciplinary contributions. This narrative begins in the late 1940s, a period when NLP, as a term, had yet to be coined, yet the foundational steps towards machine understanding of human language were being laid (Hutchins, W.J., 1986). The initial focus was on Machine Translation (MT), a task that sought to automatically translate text from one language to another, predominantly between Russian and English. This era was characterized by optimism and ambitious projects aimed at breaking down language barriers using computers.

However, the enthusiasm faced a substantial setback with the ALPAC report in 1966, which concluded that the prospects for MT were bleak and recommended a reduction in funding for such research (Hutchins, W.J., 1995). This report significantly dampened the momentum of NLP research, leading to a period of reassessment and recalibration of goals within the field. Despite this, some MT projects continued to operate, slowly refining their approaches and laying the groundwork for future successes.

The revival of NLP research in the 1980s was marked by a shift towards creating more sophisticated models of language understanding and generation. This period saw the emergence of computational linguistics as an active area of study, linking the understanding of language with logical structures and computational algorithms. Influential projects like SHRDLU, an early natural language understanding system, demonstrated the potential for computers to interpret and act upon human instructions within a controlled environment (Winograd, T., 1972). These developments highlighted the importance of syntax, semantics, and pragmatics in modeling language, leading to a deeper exploration of how linguistic principles could be encoded into computational systems.

The 1990s witnessed a paradigm shift with the introduction of statistical methods to NLP (Manning, C.D., and Schütze, H., 1999). The availability of large text corpora and advancements in computing power made it feasible to apply statistical models to language processing tasks. This approach marked a departure from rule-based systems, offering more flexibility and robustness in handling the complexities of human language. The era of statistical NLP laid the foundation for modern techniques, emphasizing the importance of data and probabilistic models in understanding language.



Today, NLP stands as a vibrant field at the intersection of AI, linguistics, and computer science, continuously evolving with the introduction of new models, datasets, and applications. Its history reflects a trajectory of overcoming challenges, embracing new methodologies, and expanding the boundaries of what machines can understand and generate in terms of human language. The future of NLP promises further integration with other disciplines, deeper understanding of nuanced linguistic phenomena, and broader applications that extend beyond current capabilities.

# Fundamentals of NLP

## Syntax vs. Semantics

Integrating discussions on syntax versus semantics within the context of Natural Language Processing (NLP) necessitates a nuanced understanding of both concepts, highlighting their distinctive roles and interconnectedness in language comprehension and generation. Drawing upon the foundational overview provided earlier and incorporating references in Harvard style, we can expand the narrative to encompass these critical linguistic components:

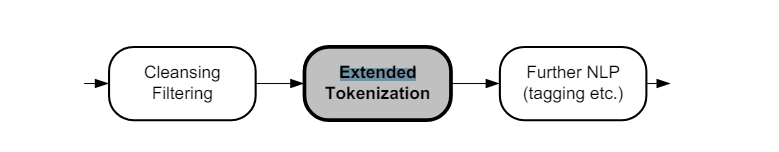
Natural Language Processing (NLP) stands at the crossroads of computer science, artificial intelligence (AI), and linguistics, aiming to bridge the gap between human communication and computational understanding. A pivotal aspect of NLP involves parsing the intricacies of syntax and semantics, each playing a unique yet complementary role in how machines interpret human language.

Syntax refers to the arrangement of words and phrases to create well-formed sentences in a language. It encompasses the rules that govern sentence structure, ensuring that the linguistic expressions are organized in a manner that adheres to the grammatical framework of the language. Syntax is crucial for NLP systems as it enables the parsing of sentences to understand their grammatical composition, which is foundational for further analysis (Chomsky, N., 1965). Without a robust syntactic analysis, machines would struggle to discern the roles of individual words and phrases within sentences, leading to potential misinterpretations of the intended messages.

Semantics, on the other hand, delves into the meaning behind words and sentences, exploring how we comprehend and attribute significance to linguistic expressions. It involves the interpretation of symbols, phrases, and sentences to understand their referential and conceptual content (Jackendoff, R., 2002). In NLP, semantic analysis allows computers to grasp the nuances of language, including idiomatic expressions, metaphors, and variations in context that influence meaning. By understanding semantics, NLP systems can go beyond mere word recognition to infer the intentions and sentiments expressed, facilitating more sophisticated interactions such as sentiment analysis, question-answering, and machine translation.

The interplay between syntax and semantics is fundamental to the NLP field. While syntax structures the form of language, semantics imbues it with meaning. Effective NLP systems leverage syntactic analysis to decode the grammatical structure of language, which, in turn, serves as a scaffold for semantic interpretation. This layered approach ensures that machines can not only recognize the formal aspects of language but also appreciate its content and context, paving the way for more nuanced and human-like understanding and generation of text (Manning, C.D., and Schütze, H., 1999).

## Tokenization

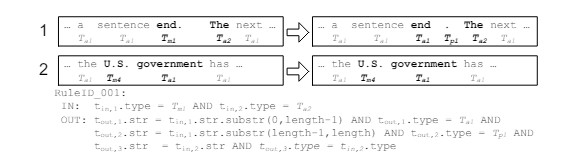
Tokenization is a fundamental process in Natural Language Processing (NLP) that involves breaking down text into smaller units, such as words, phrases, or symbols, known as tokens. This process is crucial for preparing raw text for further NLP tasks such as parsing, part of speech tagging, and semantic analysis. Tokenization simplifies

the complex structure of text, making it more manageable for algorithms to process and analyze.

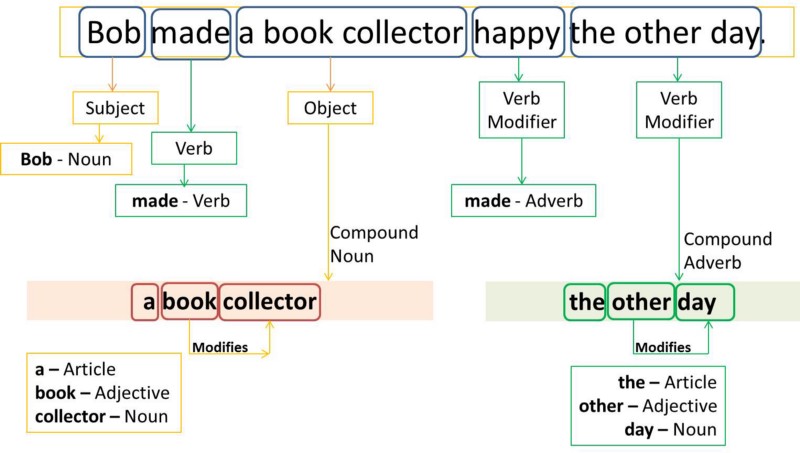
The significance of tokenization lies in its ability to delineate the boundaries between elements in a text, which are essential for understanding the grammatical and semantic

structure of the language (Jurafsky, D., and Martin, J.H., 2019). By segmenting text into tokens, NLP systems can more effectively apply linguistic rules and perform analyses that contribute to tasks like sentiment analysis, machine translation, and information retrieval.

There are several approaches to tokenization, including simple white-space tokenization, where tokens are defined by spaces, and more sophisticated methods that consider punctuation, special characters, and multi-word expressions. Advanced tokenization techniques may also involve linguistic analysis to correctly identify tokens in complex scenarios, such as distinguishing between the period in abbreviations versus the end of a sentence (Manning, C.D., and Schütze, H., 1999).

Tokenization serves as the first step in preprocessing text data, setting the foundation for subsequent NLP tasks. It is essential for ensuring that the input text is in a structured form that computational models can interpret and analyze efficiently. By breaking down text into manageable pieces, tokenization enhances the ability of NLP systems to perform a wide range of linguistic and semantic analyses, contributing to the overall effectiveness of language processing applications (Bird, S., Klein, E., and Loper, E., 2009).

## Part-of-Speech Tagging

Part-of-Speech (POS) Tagging is a critical preprocessing step in Natural Language Processing (NLP) that involves assigning parts of speech to each word in a given text, such as nouns, verbs, adjectives, etc., based on both its definition and its context. This process is fundamental for understanding the grammatical structure of sentences and is instrumental in various NLP tasks including parsing, named entity recognition, and even in machine translation and sentiment analysis.

The importance of POS tagging stems from its ability to contribute to the syntactic analysis of texts, thereby facilitating a deeper understanding of the meanings behind sentences. By categorizing words into their respective parts of speech, NLP systems can apply grammatical rules and perform more complex analyses, such as detecting relationships between words and understanding sentence structure (Jurafsky, D., and Martin, J.H., 2009).

Modern POS tagging utilizes machine learning algorithms, including hidden Markov models, maximum entropy models, and more recently, deep learning approaches that can take advantage of large annotated corpora for training. These models have significantly improved the accuracy of POS tagging, making it a reliable process in automated text analysis (Manning, C.D., 2011).

The application of POS tagging extends beyond simple grammatical analysis. It plays a crucial role in improving the performance of NLP tasks by providing additional contextual information. For instance, in sentiment analysis, knowing the parts of speech can help in accurately identifying opinion words. Similarly, in machine translation, understanding the grammatical role of words in sentences can enhance translation quality by ensuring that words are correctly interpreted and translated according to their usage (Smith, N.A., 2011).

## Name Entity Recognition

Named Entity Recognition (NER) is a crucial task in Natural Language Processing (NLP) that involves identifying and classifying named entities within a text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. NER is fundamental for extracting structured information from unstructured text, making it essential for numerous applications like information retrieval, question answering systems, content classification, and knowledge graph construction.

The significance of NER lies in its ability to discern and categorize key elements in text, thereby enabling more effective organization and retrieval of information. This process aids in the understanding of the context and the relationships between entities, which is critical for semantic web applications, summarization, and enhancing search algorithms (Nadeau, D., and Sekine, S., 2007).

Advancements in NER have been driven by machine learning and deep learning techniques, utilizing both supervised and unsupervised learning models. The advent of recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, and more recently, transformer-based models like BERT (Bidirectional Encoder Representations from Transformers), have significantly improved the accuracy and efficiency of NER systems (Lample, G. et al., 2016; Devlin, J. et al., 2019).

These technological advancements have expanded the potential of NER in processing vast amounts of text data across various domains, from news articles and scientific papers to social media posts. By extracting meaningful entities, NER systems contribute to the automation of document analysis, enhancing information extraction processes and enabling more sophisticated data analysis and decision-making processes.

## Dependency Parsing

Dependency Parsing is a vital process in Natural Language Processing (NLP) that focuses on identifying the grammatical structure of sentences by establishing relationships between "head" words and words that modify those heads. This approach highlights how words in a sentence depend on each other for meaning, thereby mapping out the sentence's grammatical structure in terms of dependencies between words. Unlike traditional phrase structure parsing, which constructs a hierarchical tree structure of sentences, dependency parsing builds a tree that represents the dependencies between individual words, making it crucial for understanding the syntactic and semantic relationships within a sentence.

The importance of dependency parsing in NLP cannot be overstated. It enables the analysis of the syntactic structure of sentences, which is essential for numerous advanced NLP tasks, including machine translation, sentiment analysis, information extraction, and question answering. By understanding the grammatical relationships between words, NLP systems can interpret the context and meaning of sentences more accurately, leading to more effective processing of natural language texts (Kübler, S., McDonald, R., and Nivre, J., 2009).

Recent advances in dependency parsing have been driven by machine learning models, especially with the advent of deep learning techniques. These models, particularly those based on neural networks, have significantly improved the accuracy and speed of dependency parsing, enabling the processing of complex languages and intricate sentence structures (Chen, D., and Manning, C.D., 2014). Furthermore, the development of universal dependency treebanks and the use of transformer-based models like BERT (Devlin, J. et al., 2019) have contributed to the cross-linguistic applicability and performance of dependency parsers.

The application of dependency parsing extends beyond linguistic analysis to practical applications such as improving the efficiency of search engines, enhancing the relevance of search results through better understanding of query intent, and automating the summary generation of texts by identifying key syntactic elements.

## Lemmatization and Stemming

Lemmatization and stemming are two text normalization techniques used in Natural Language Processing (NLP) to prepare texts, documents, or words for further processing. Both methods are aimed at reducing the inflectional forms of words to their root form, but they operate differently and serve distinct purposes.

### Lemmatization

Lemmatization is the process of reducing a word to its base or dictionary form, known as the lemma. Unlike stemming, lemmatization considers the context and uses a full vocabulary of a language to apply morphological analysis to words. The lemma of a word is its canonical form or the form you would find in a dictionary. For instance, "running", "ran", and "runs" are all lemmatized to "run". Lemmatization is particularly useful in tasks that require high levels of accuracy and context understanding, such as semantic indexing, text analysis, and comprehensive information retrieval (Manning, C.D., Raghavan, P., and Schütze, H., 2008).

### Stemming

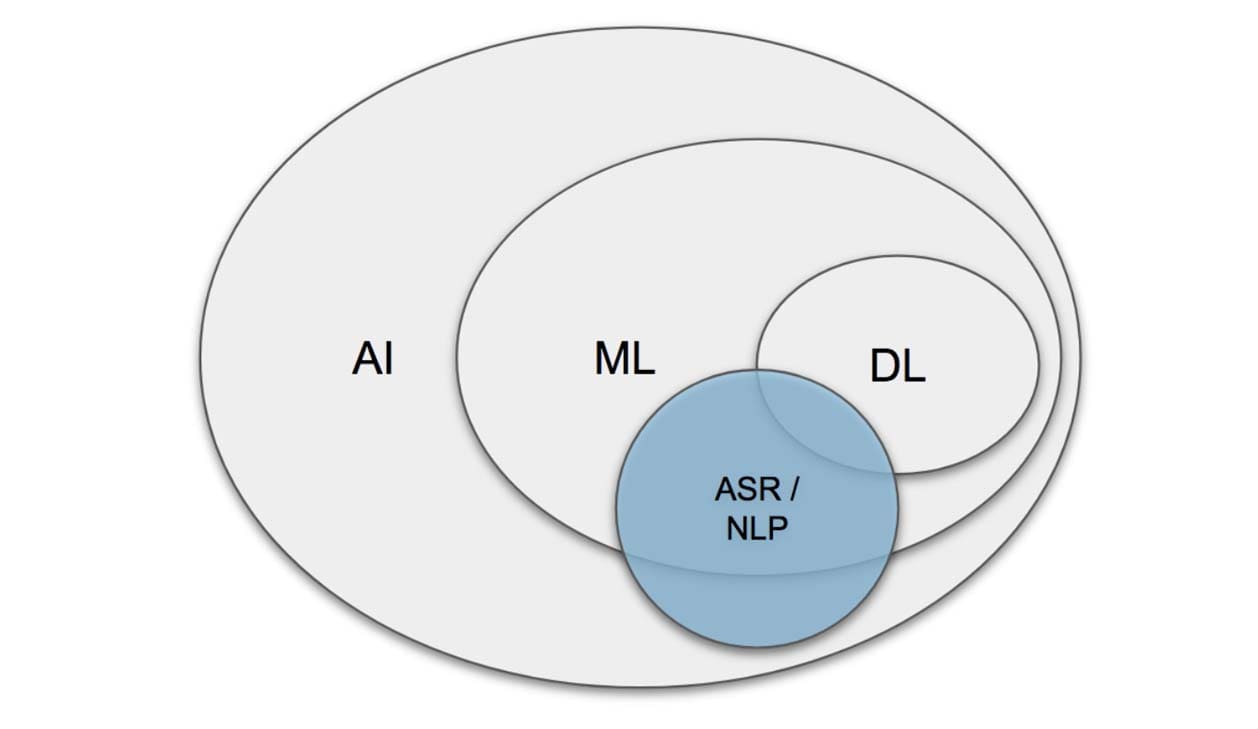
Stemming, on the other hand, is a heuristic process that strips the suffixes from words, aiming to achieve this reduction more aggressively than lemmatization. The objective is to reduce words to their word stem, base or root form—often a part of the word that is not itself a valid word in the language. For example, "fishing", "fished", "fisher" all reduce to the stem "fish". Stemming algorithms, such as the Porter stemmer, are simpler and faster than lemmatization, making them suitable for search queries and systems where the broad recall is more important than precise accuracy (Porter, M.F., 1980).

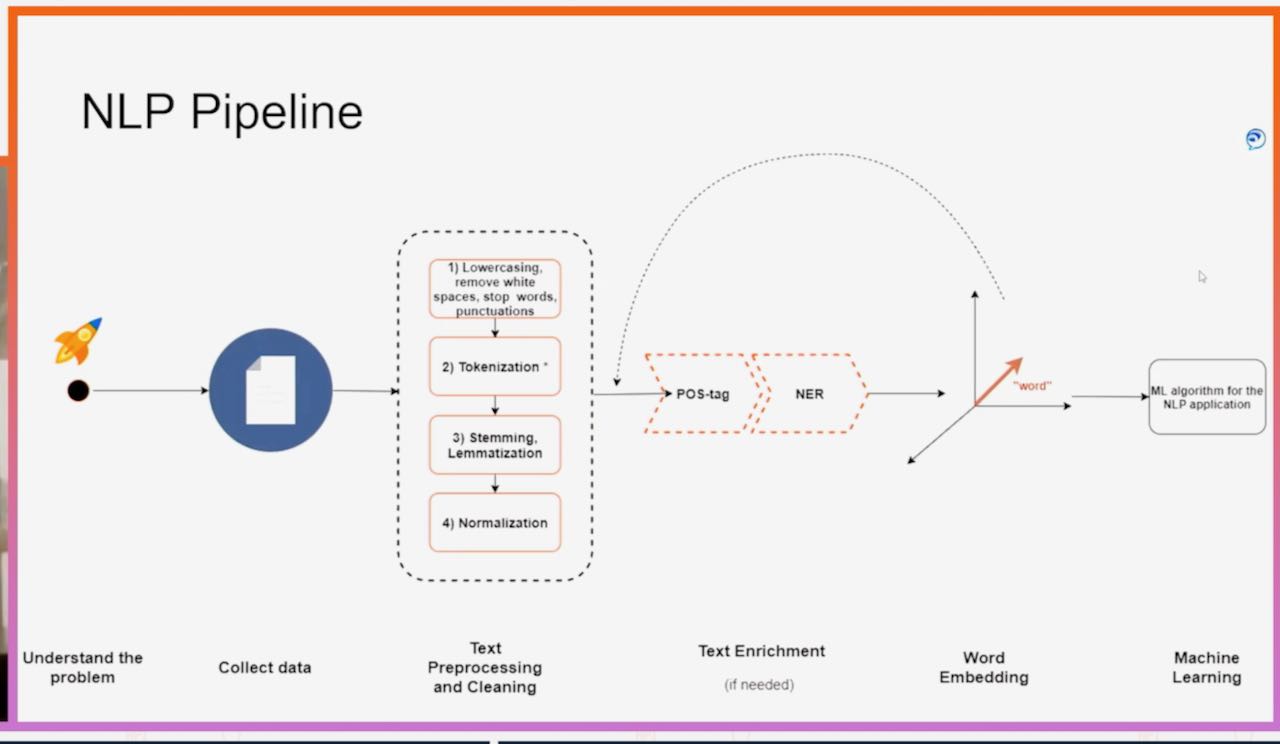
### Comparative Analysis

While both lemmatization and stemming aim to condense words to their root forms, lemmatization does so with an understanding of the morphological analysis of words, making it more accurate but computationally expensive compared to stemming. Stemming algorithms, due to their heuristic nature, are faster but can generate non-words and are less accurate in understanding the context of the word in a sentence.

## Machine learning in NLP

Machine Learning (ML) plays a pivotal role in Natural Language Processing (NLP), driving advancements that have transformed how machines understand, interpret, and generate human language. By leveraging patterns in data, machine learning algorithms enable computers to perform complex NLP tasks without explicit programming for each specific task. This integration of ML in NLP has facilitated the development of applications such as speech recognition, sentiment analysis, machine translation, and chatbots, among others.

Machine learning models, ranging from traditional algorithms like Naive Bayes, decision trees, and support vector machines (SVMs) to advanced neural network architectures, have been instrumental in NLP. These models are trained on large datasets, learning to predict or classify text data based

The advent of transformer models, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pretrained Transformer), represents a significant leap forward. These models have set new standards for a wide range of NLP tasks by effectively capturing deep contextual relationships within text (Vaswani, A. et al., 2017; Devlin, J. et al., 2019). Their ability to pre-train on vast amounts of text data and fine-tune for specific NLP tasks has led to unprecedented accuracy in language understanding and generation.

Despite these advances, integrating machine learning into NLP poses challenges, including the need for large annotated datasets for training, the complexity of understanding language nuances, and the computational resources required for training sophisticated models. Additionally, there's an ongoing effort to improve model interpretability and reduce bias in machine learning-based NLP systems.

# The Role of NLP in Enhancing Access to Justice

The role of Natural Language Processing (NLP) in enhancing access to justice is increasingly significant, leveraging the power of AI to bridge the gap between legal services and those in need. By automating and streamlining legal processes, NLP technologies make legal information more accessible, understandable, and actionable for individuals and communities who may otherwise lack the resources for traditional legal representation.

**Demystifying Legal Language**

One of the primary barriers to accessing justice is the complexity of legal language. NLP tools can translate legal jargon into plain language, making legal documents, regulations, and procedures more understandable to the layperson. By providing summaries of legal texts and explaining legal concepts in simple terms, NLP aids individuals in navigating the legal system more confidently and making informed decisions (Katz, D.M., Bommarito II, M.J., and Blackman, J., 2017).

**Automated Legal Assistance**

Legal chatbots and virtual assistants, powered by NLP, offer preliminary legal advice and assist in document preparation, ranging from simple contracts to more complex legal filings. These applications can guide users through legal processes, ask relevant questions, and generate documents based on the user's responses, significantly lowering the barrier to initiating legal actions or responding to legal issues (Sourdin, T., 2018).

**Enhancing Legal Research**

NLP facilitates advanced legal research by enabling the efficient analysis of vast amounts of legal texts, case law, and legislation. Lawyers and legal researchers can use NLP-powered tools to quickly find relevant precedents, identify legal trends, and gather evidence to support their cases. This not only improves the quality of legal representation but also makes it more cost-effective, indirectly benefiting clients with limited resources (Alarie, B., Niblett, A., and Yoon, A.H., 2018).

**Accessible Dispute Resolution**

Online Dispute Resolution (ODR) platforms utilize NLP to offer accessible means for resolving disputes outside of traditional court settings. By automating parts of the mediation or arbitration process, ODR platforms can resolve conflicts more quickly and with less financial strain on the parties involved, making justice more accessible for all (Rule, C., 2017).

# Identifying and Mitigating Bias

## Privacy and Security Concerns in Legal NLP Applications

Privacy and security concerns are paramount in the deployment of Natural Language Processing (NLP) applications within the legal domain. Legal NLP applications, which handle sensitive and potentially confidential information, necessitate stringent measures to protect client data and ensure compliance with legal standards and regulations. The processing of legal documents, client communications, and other sensitive information by NLP systems raises substantial concerns about data protection, unauthorized access, and the potential misuse of information.

One of the primary challenges in legal NLP applications is ensuring the confidentiality and integrity of the data processed. Legal documents often contain privileged information, trade secrets, and personal data subject to various privacy laws, such as the General Data Protection Regulation (GDPR) in Europe (Voigt, P., and Von dem Bussche, A., 2017). As such, NLP systems used in the legal field must incorporate robust encryption methods for data storage and transmission, alongside secure access controls to prevent unauthorized access to sensitive information.

Furthermore, the use of NLP in legal applications involves ethical considerations regarding the fairness and transparency of automated systems. Bias in NLP models can lead to unfair outcomes or discrimination, undermining the trust in automated legal analyses (Barocas, S., Hardt, M., and Narayanan, A., 2019). Therefore, legal NLP applications must be designed with fairness in mind, incorporating methods to detect and mitigate bias in training data and model predictions.

Another significant concern is the potential for data breaches and the unauthorized disclosure of sensitive information. Legal NLP applications must comply with legal standards for data protection, implementing stringent security protocols and regularly auditing systems for vulnerabilities (Romanosky, S., 2016). Moreover, there is a need for clear guidelines and regulations governing the use of AI and NLP in legal contexts, ensuring that these technologies are used responsibly and ethically.

In summary, privacy and security concerns in legal NLP applications are critical issues that require careful consideration and proactive measures. Protecting sensitive legal information while ensuring the fairness and transparency of NLP systems is essential for maintaining client trust and compliance with legal and ethical standards.

## Current Trends and Future Directions of NLP in law

The integration of Natural Language Processing (NLP) within the legal domain has been transformative, reshaping how legal professionals interact with vast amounts of textual data and streamlining various aspects of legal research, document analysis, and client services. As NLP technologies continue to evolve, several current trends and future directions are emerging, promising to further revolutionize the practice of law.

**Current Trends**

**1**. ***Automated Legal Document Analysis:***

The use of NLP for automating the analysis of legal documents, including contracts, court opinions, and legislation, has become increasingly sophisticated. Tools powered by NLP algorithms can now identify, extract, and summarize relevant information from legal texts, significantly reducing the time and effort required for legal research and due diligence (Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., and Sun, M., 2020).

**2**. ***Legal Chatbots and Virtual Assistants:***

Legal chatbots and virtual assistants, equipped with NLP capabilities, are becoming more prevalent. These technologies offer legal advice to the public, help in drafting simple legal documents, and provide support for customer service operations in law firms, making legal services more accessible (Kreutzer, R.T., and Sirrenberg, M., 2019).

**Future Directions**

**1. *Enhanced Legal Predictive Analytics:***

The future of NLP in law includes the development of more advanced predictive analytics tools. By analyzing historical legal data, NLP models could predict the outcomes of cases, helping lawyers make better-informed decisions about case strategies and likelihood of success (Ashley, K.D., 2017).

**2.** ***Ethical and Fair Use of NLP in Legal Applications:***

As NLP technologies become more embedded in legal processes, there will be an increased focus on ensuring the ethical use of these tools. This includes addressing concerns related to bias, transparency, and the explainability of NLP models to ensure fair and equitable legal outcomes (Branting, L.K., 2021).

**3. *Cross-lingual and Multijurisdictional Legal NLP Applications:***

Future NLP systems will likely become more adept at handling multiple languages and legal jurisdictions, facilitating cross-border legal research and global compliance tasks. This advancement could significantly benefit international law firms and organizations dealing with multinational legal issues (Tsarapatsanis, D., and Aletras, N., 2021).

# Project Proposal

## Project Details:

Litigate would be an online, hosted chat application accessible on mobile phones and computers. Users can find legal advice related to household and tenant law on the go. There will be a minimum age limit for accessing the platform, but people from all backgrounds will be able to use the application. An NLP model working in the backend, in relation to a database, will process user queries in real-time and produce appropriate responses using NLP techniques. The responses will include basic advice about the matter and, if available, a case law relevant to the issue from the database

## Project Specification:

**Project Scope and Objectives:**

The scope of the project has been defined by the identified said objectives that need to be accomplished throughout the development of the project. The primary objectives carry more weight as they would be allocated the most resources and time before the presentation of the project in April,2024. If the primary objectives are achieved only then the secondary objectives would be considered implementing into the overall workflow of the application.

**Primary Objectives:**

1. ***User Interface Development:*** Design a user-friendly interface for both mobile and desktop platforms that allows users to easily interact with the chat application.
2. ***Natural Language Processing Integration:*** Implement a robust NLP model to understand and process user queries accurately in real-time.
3. ***Legal Database Creation:*** Compile a comprehensive database that includes relevant case laws, statutes, and legal precedents pertaining to household and tenant law.
4. ***Real-time Response Generation:*** Develop a system capable of generating accurate legal advice and relevant case law references in response to user queries.

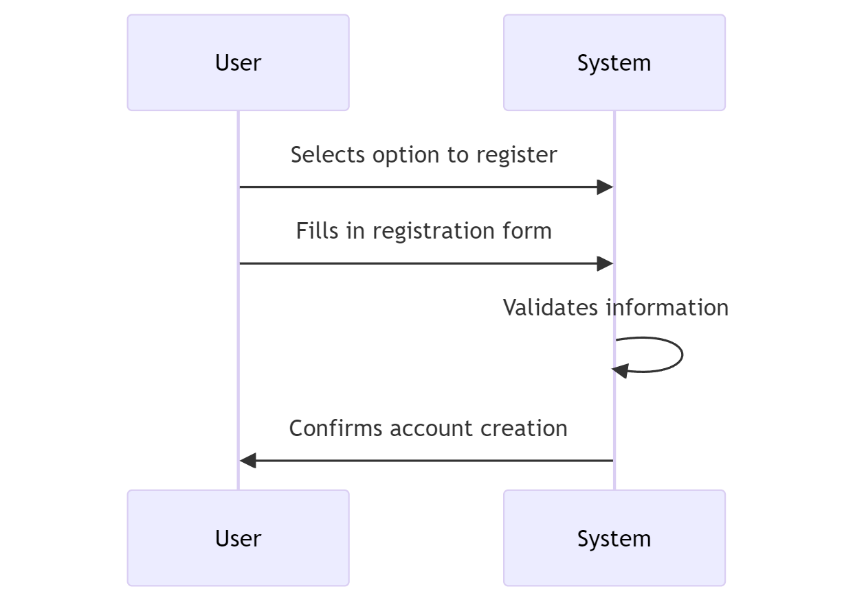
**Secondary Objectives:**

1. ***Accessibility and Inclusivity*:** Ensure the app is accessible to users from all backgrounds, with considerations for those with disabilities
2. ***Security and Privacy:*** Implement robust security measures to protect user data and ensure privacy, especially when handling sensitive legal queries.
3. ***User Authentication and Age Verification:*** Create a secure user authentication system with age verification to enforce the minimum age requirement for accessing the platform.
4. ***Feedback Mechanism:*** Incorporate a feedback mechanism to collect user responses on the accuracy and helpfulness of the legal advice provided, facilitating continuous improvement.

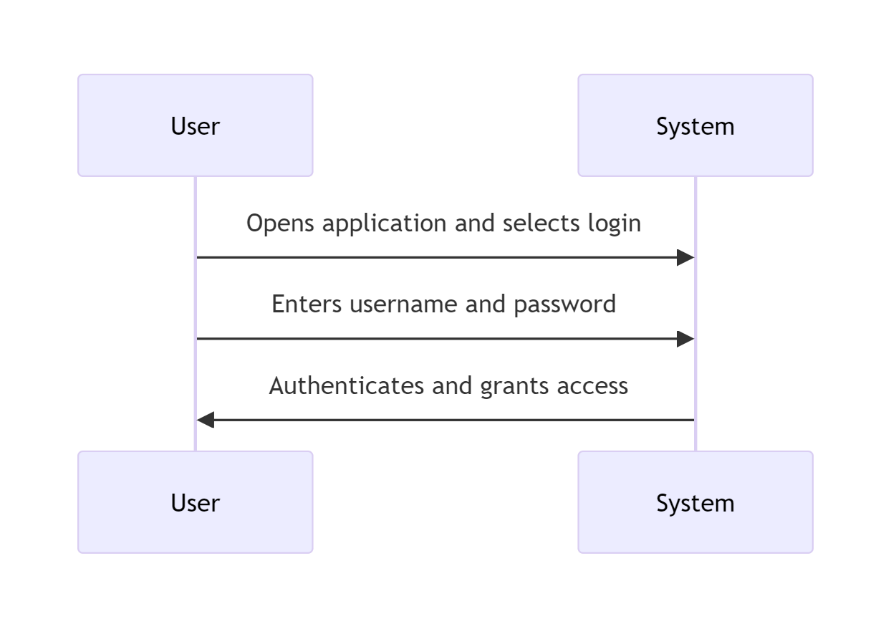
## Project User Stories

Keeping the Objectives and the idea of the project in mind. I have brainstormed the potential use cases a user can have with my application. Which would further be broken down using other analysis techniques such as MoSCoW and depending on the resources and time available.

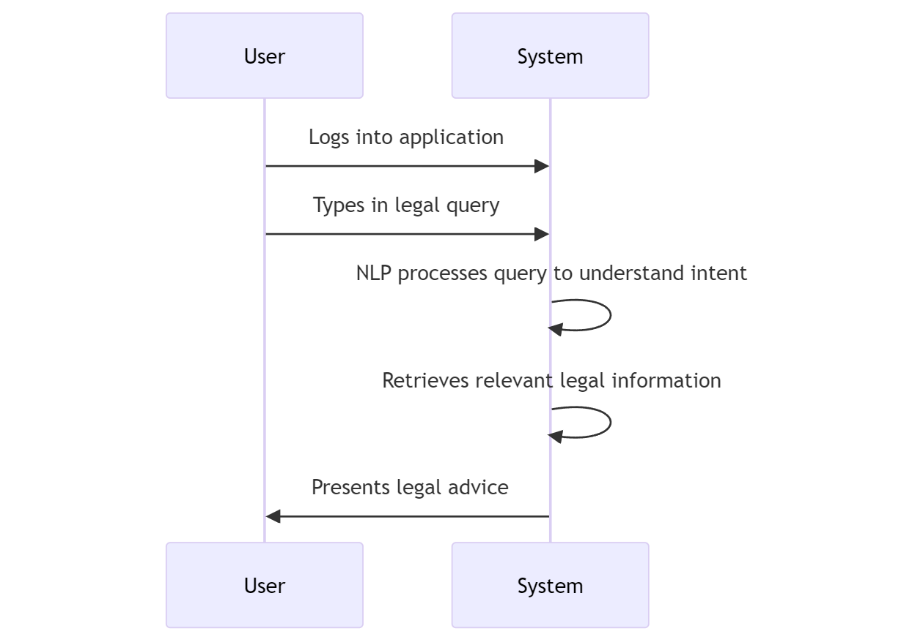
**Use Case 1:** User Registration



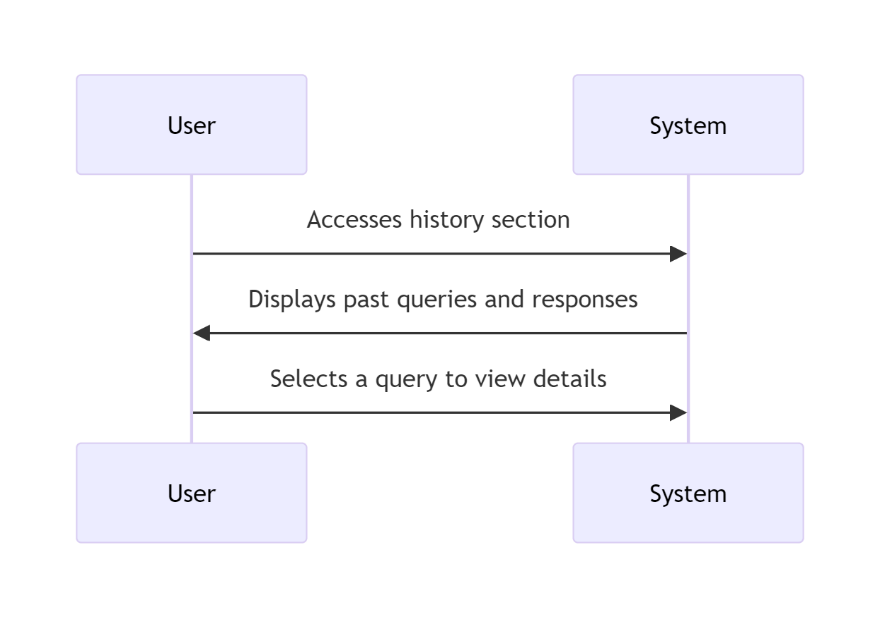
**Use Case 2:** User Login



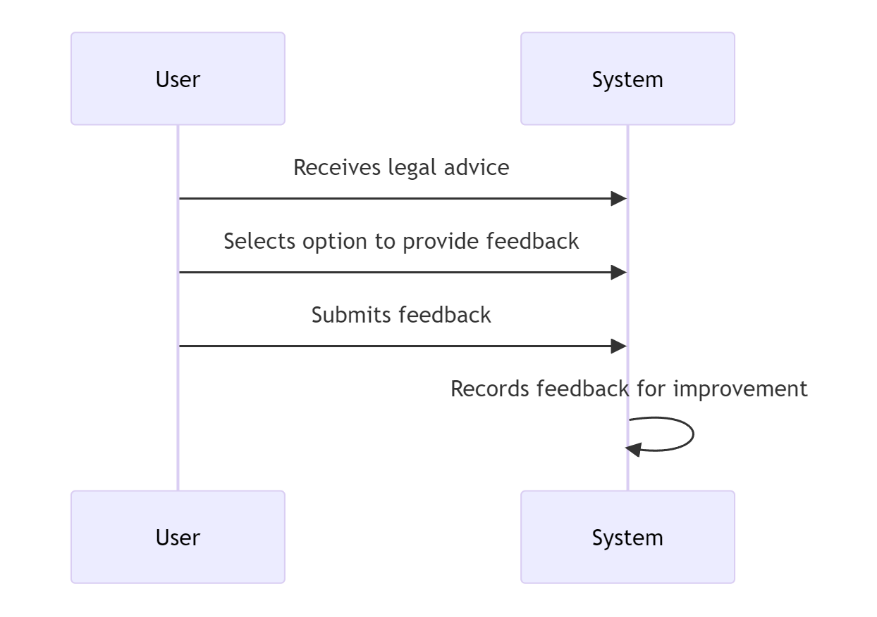
**Use Case 3:** User Receives Legal Advice



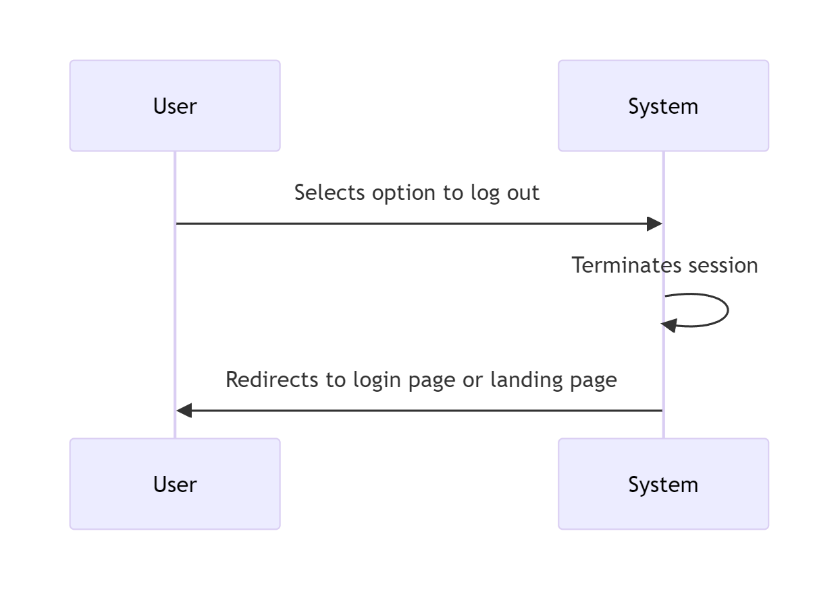
**Use Case 4:** User Reviews Chat History



**Use Case 5:** User Provides Feedback



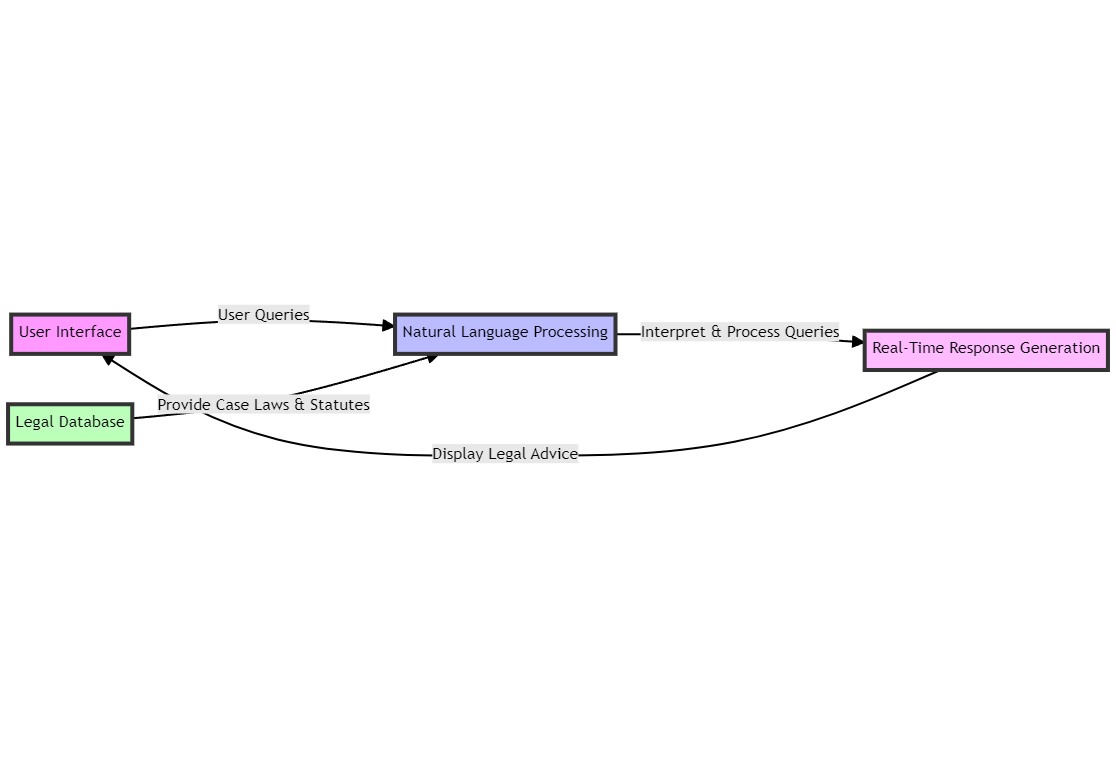
**Use Case 6:** User Logs Out



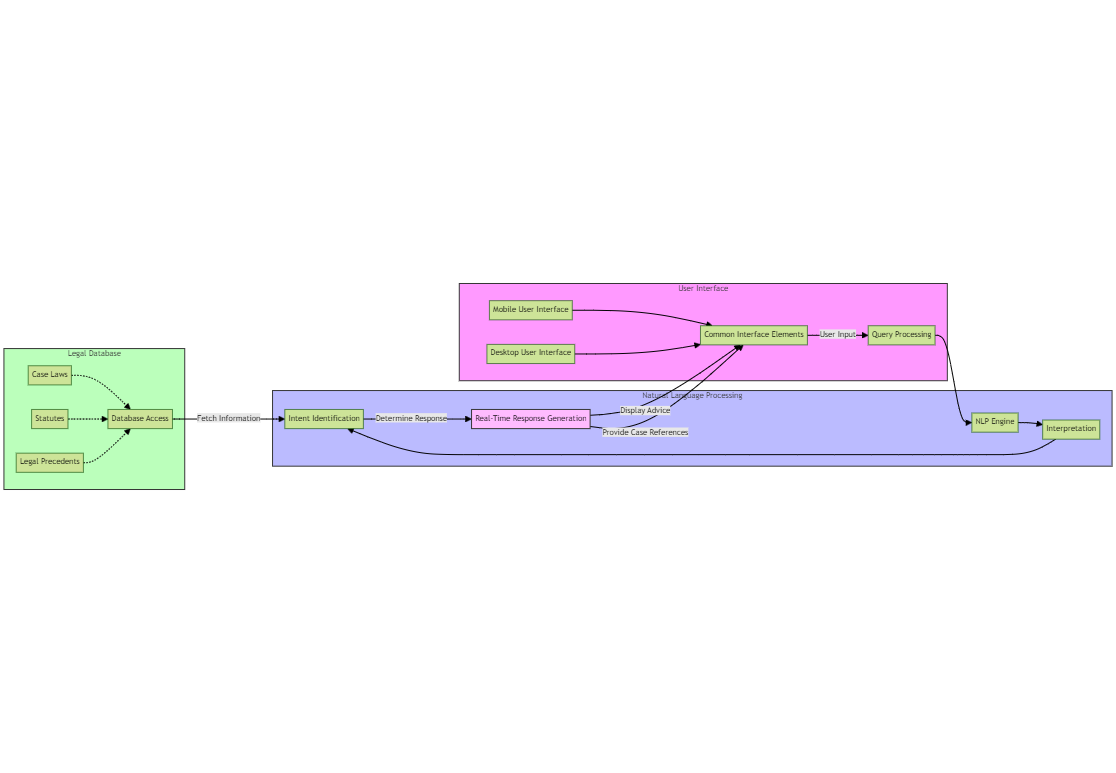
# High Level System Design:

In line with the user stories I've decided to focus on, sketching out a high-level design (HLD) helps us map out the system's architecture in a nutshell. This overview helps pinpoint the necessary hardware and software interactions, data exchanges, and communication flows across the system. At the heart of the architecture is a user interface designed to retrieve and update information from database tables. One of the key advantages of our approach is its scalability. As new data needs arise in the future, our system can easily expand to accommodate these additional data sources.

This diagram shows a high-level flow of data between different software and hardware components.



One of the disadvantages of using the architectural design and solution, is the current technological skillset that is needed to achieve this implementation of the project. The Mitigation of the potential risks will be taken into account by discussing the it with the supervisor every week.



The diagram presented offers a detailed view of our proposed high-level design, showcasing the different elements, connections, and data movements.

# Requirement Analysis:

The requirements would be defined using the MoSCoW Analysis and according to the objectives defined before.

## MoSCoW Analysis

Based on the objectives mentioned in the Project Scope and objective I was able to shortlist the requirements for the project using the Moscow Method. Where the Must have requirements take the highest priority followed by should have and could have requirements respectively. And I thought of the requirements that could be implemented as a part of future development of the project

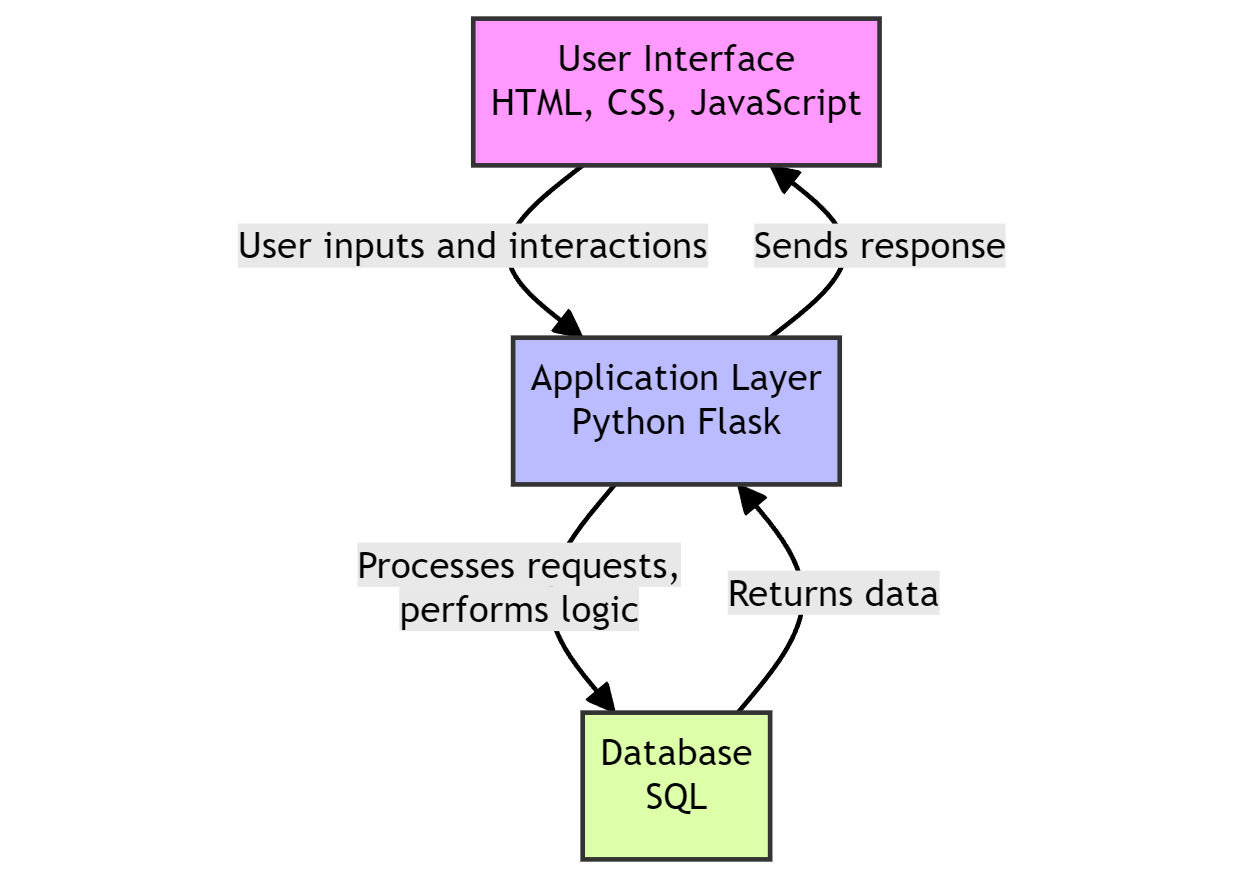
|  |  |  |
| --- | --- | --- |
| **Priority** | **System Component** | **Details** |
| Must Have | User Account Management | User registration with data validation. |
|  |  | Secure user authentication. |
|  |  | User session management and secure logout. |
|  |  | End-to-end encryption of user data. |
| Must Have | Core System Functionality | Integration of NLP engine for natural language processing. |
|  |  | Real-time legal advice generation based on NLP. |
|  |  | Legal database connectivity for dynamic access. |
|  |  | Automated retrieval and presentation of case laws and statutes. |
| Should Have | User Interface | Responsive design for mobile and desktop platforms. |
|  | Experience | Feature for saving and displaying user query history. |
|  |  | User-friendly FAQ section for common inquiries. |
| Could Have | Extended Functionality | Multilingual support for diverse user base. |
|  |  | Notification system to alert users about updates or responses. |
| Won't Have | Future Considerations | Hosting on a real domain for processing real-time queries |
| (This Time) |  | Advanced document generation |

## Project System Requirement:

Based on the requirement analysis done for the project. I have decided on the software and hardware requirements of the project. As the requirements of the project are modest and it would only be hosted on the localhost for the MVP. I have kept the requirements at minimum for the functioning of the project.

**Development Languages and Tools**:

1. ***Backend Programming Language:*** Python, chosen for its extensive library support, particularly for natural language processing (NLP) tasks. I decided to choose this language for the access of the existing information for python in compared to other languages such as C++. And another factor that contributed to its selection was my own proficiency in the language given the duration of the project. It would have been very highly unlikely to complete the project if I had decided to proceed with the language, I didn’t have command on.
2. ***Frontend Web Technologies:*** HTML, CSS, and JavaScript, for creating an interactive and user-friendly web interface. The User interface is decided to kept simple as most of the resources and time would be allocated to developing the NLP model. Hence, no frontend framework is being used at the moment but if time is left. I’m planning to implement React as the frontend framework for the application.
3. ***NLP Library:*** NLTK, utilized for implementing the chat application's NLP capabilities to process and understand user queries. For developing and training the model I’m going to be using Torch library. Which would be used in the definition of neural networks and processing the input from the user. From the NLTK libraries, I’m going to be using PortStemmer to implement functions such as tokenization, stemming and bag of words.
4. ***Web Framework:*** Flask, selected for its simplicity and efficiency in setting up lightweight web applications, suitable for a project focused on functionality demonstration. Option of Django was considered as well for the development of the project but given the simple nature of the MVP I decided to settle down for flask. And I had some previous experience using this framework so it was a natural choice for me.
5. ***Database System:*** SQLite or Firebase, due to its ease of integration with Python applications and sufficiency for the storage needs of a development-stage project.
6. **Integrated Development Environment(IDE):** VsCode, recommended for its support for Python and web technologies, alongside integrated version control.

****

**Hardware Requirements:**

Given the localized hosting and demonstration focus of the project, the hardware requirements are kept simple:

1. ***Processor***: A minimum of a Dual-core processor is required to ensure smooth running of the development server and NLP processes.
2. ***Memory:*** At least 4 GB RAM to support the simultaneous execution of the development environment, web server, and any ancillary tools.
3. ***Storage:*** A minimum of 5 GB of available disk space to accommodate the application codebase, SQLite database, and dependencies.

# Project Solution

## Assumptions, Constraints, Risks:

**Assumptions:**

***Technical Capabilities:***

It's assumed that the chosen technologies and frameworks for NLP(NLTK), machine learning (Torch), and other functionalities will effectively support the development of Litigat8.As this is a realm still to be explored and considering the limited time and resources available full-scale deployment of the project can be somewhat troublesome. But the given requirements for a Minimum Viable Product (MVP). It is assumed that all the technical requirements are satisfactory.

***Data Availability:***

Adequate and accurate tenant-landlord law documents, statutes, and regulations are available for populating the database. A I’m going to be preparing the dataset myself. It is assumed that the dataset would have adequate information for the model to produce appropriate responses for the user. And all the information would contain in the dataset is correct. Even though there can be discrepancies in the dataset

***User Feedback:***

Users will provide feedback for system enhancement and that the feedback will be representative and constructive. It is assumed that the feedback provided is constructive and explainable but still there can be instances where the feedback isn’t constructive or comprehensible. And User Feedback is one of the secondary objectives hence it would take less resources during the production of software.

***Compliance and Legal Clearance:***

All necessary legal clearances and compliance requirements related to providing legal advice and handling user data will be obtained. It is assumed that the model has been approved of giving legal advices in the matters of household and tenant law. As law and AI is a very tricky domain where you have to be careful about the data that is produced and dispersed to the user. As this is a demonstration of its capabilities and the project is being made for ONLY this purpose. Hence, it is assumed it has all the legal clearance

***Budget and Time:***

The project will be completed within the allocated budget and time frame. According to the defined user stories and analysis of requirements using methods like MoSCoW. It is assumed that all the set requirements would be completed within Budget and Time. But as this can vary depending on various factors. But for the sake of convivence an assumption is put into place.

**Constraints:**

***Single Stakeholder:***

As the sole stakeholder, your availability and decision-making will be critical for the project's progress and direction. As I’m going to be the only one that would be directing the progress of the project along with the supervisor which can put a bit of constraint on the project in a sense that I would have to make all the critical decision instead of having a shared responsibility of the decision-making progress.

***Budget Limitations:***

Budget constraints may limit the scope of development, particularly concerning the scalability and future plans for web and mobile interface accessibility. The options of cloud hosting and own professional domain will be looked in the future if there are enough resources allocated to the project if there are investors involved in the project but for this development cycle it is constraint to be hosted locally without a domain.

***Technological Limitations:***

The performance of Litigat8 may be constrained by the capabilities of the chosen technologies and frameworks. Some of the technologies chosen for the project for instance python in itself may not be the best choice when it comes to making heavy load models which are capable of processing natural language. But for the sake of demonstration and to have an MVP to show for by the end of April. Some of the technologies were given more weight.

***Geographical and Jurisdictional Limitations:***

Legal advice and statutes provided will be limited to specific geographical regions or jurisdictions. As this model is being developed just for the jurisdiction of UK law hence it would only be able to provide support and advice for Tenant and Household law in UK.

**Risks:**

Some of the foreseeable risk have been identified that could potentially happen in the duration of the project. The mitigation technigques for these risks would be discussed with the supervisor in the weekly meetings. And sufficient alternative pathways would be setup for the project if I encounter any unforeseen circumstances which could lead to failure of any component of the project.

***Data Privacy and Security:***

Risks associated with user data privacy and security breaches. As all the data shared by the user would be of confidential nature because of it being associated with their private matters regarding household and tenant law. Hence, there are risk of data breach in a sense if a third-party gets access to the history of the user chats and may have a malignant intent. Thus for the mitigation of this risk sufficient steps would be taken to ensure the data is kept safe.

***Technology Failure:***

Risks of technology failure impacting the accuracy and reliability of legal advice provided. As there are going to be lots of functioning elements in the project from the interface to the database lots of different factors would be involved. Hence, there is a risk of technological failure if one of the components crashes or fails due to unforeseen circumstances. But sufficient steps would be taken in order to prevent that from happening. Ensuring good coding practices and proper documentation of the project to ensure any unforeseen circumstances should be delt with using proper protocols put in place.

***Dependency on External Systems:***

Future dependency on external cloud hosting platforms and cloud-based database like firebase. I have yet to decided if I want to go for a inbuild database like SQLite or Firebase. If I decided on settling for firebase then there is a potential risk that the application might not work because of being depended on an external firebase database. But steps would be taken and would be discussed with the supervisor to mitigate the risk as much as possible.

## Solution Description:

Based on the extensive information that was collected during the research and the requirement analysis of the project, sufficient evidence is provided for the case that a web application would be suffice to resolve the said user stories and objectives that have been identified for the project. A web application would enable to design a proper interface for the user for ease of access along with a functioning backend layer with an NLP model which in itself is very intensive and alongside that would the database for the whole system which would allow to create, update and delete user data along with the chat data. However, due to number of limitations, it is limited in its ability to create complete system. Therefore, the proposed system would deliver a proof of concept, which would only show the potential of the system in case if it’s implemented as a complete system.

The poof-of-concept application (litigat8) will showcase the process to input the query as a string and then the system would be able to process the information to come with appropriate responses along with case laws and statues if they exist for that specific case

The application would showcase how the user would be able to go back to his/her queries if he or she wants to look over his/her queries again.

# Solution Delivery Approach:

## Project Management Approach:

**Agile Methodology**

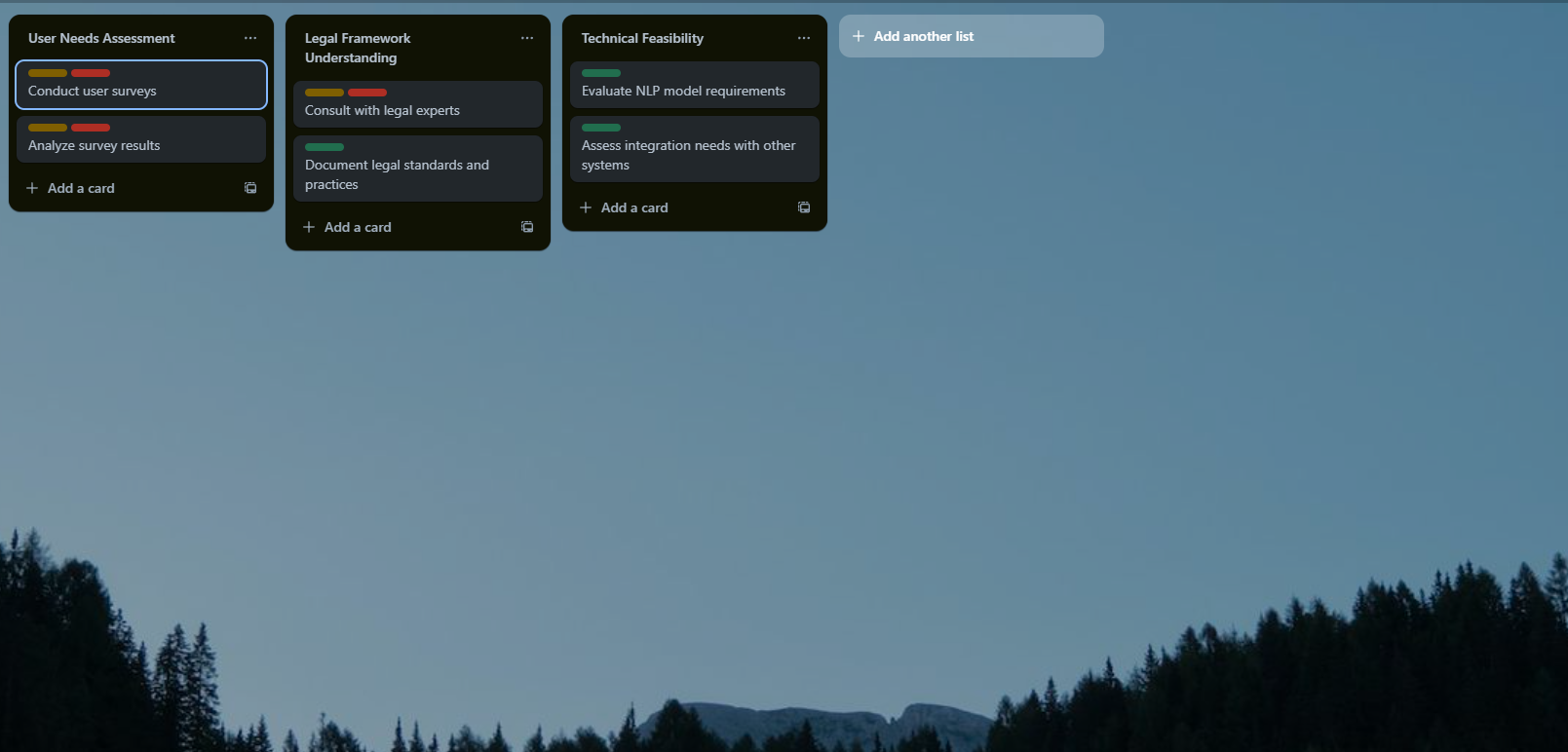
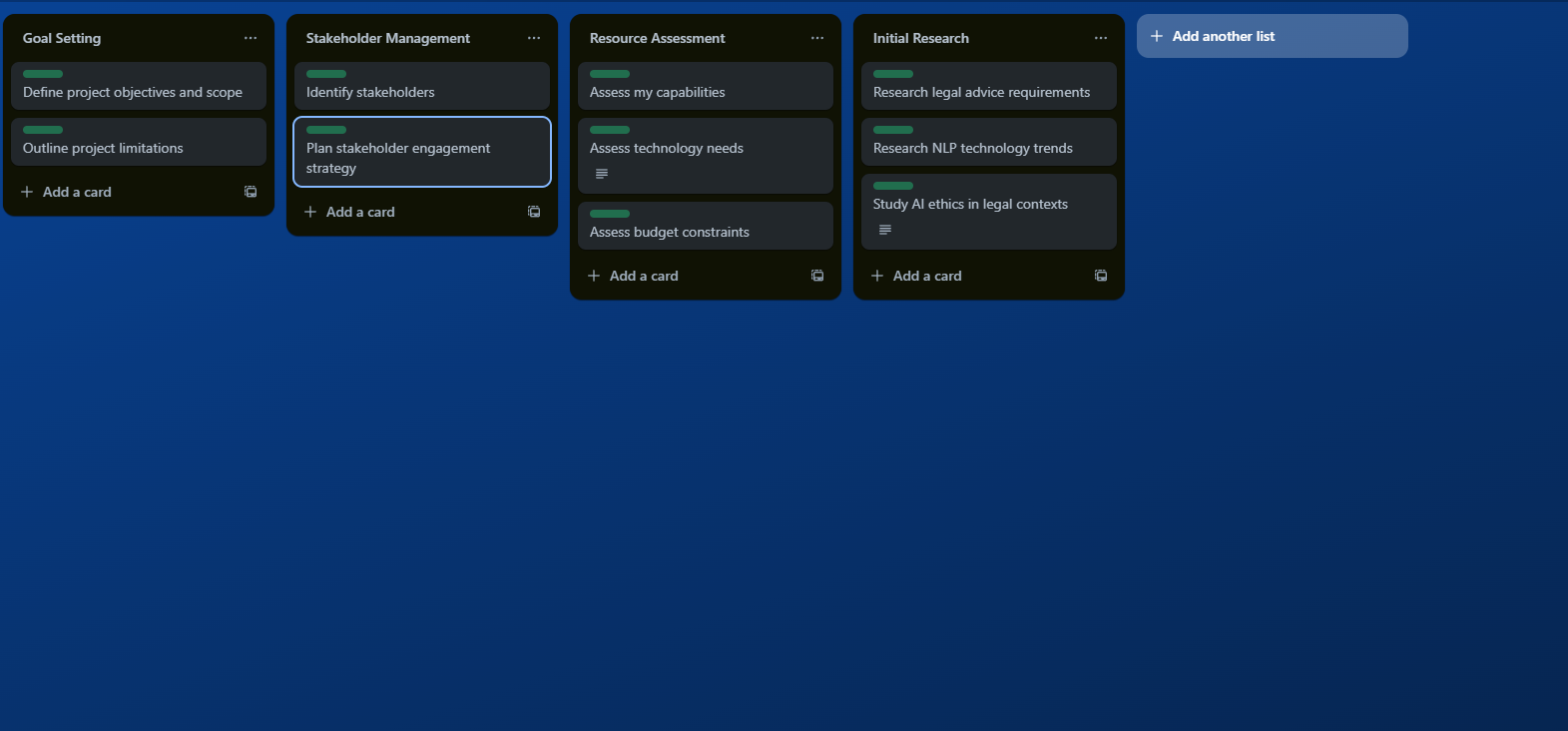
In line with modern software development practices, the project adopted the Agile methodology starting from October 15, with a projected end date of April 15. This approach, characterized by its flexibility and iterative nature, emphasizes adaptability and responsiveness to evolving requirements. Agile methodology, as defined by Beck et al. (2001), offers a dynamic alternative to the rigid, linear progression of the Waterfall model. It is conducive to environments like ours, where user needs and system functionalities may not be fully delineated from the onset.

**Project Management Using Scrum Framework**

Scrum, a subset of Agile, was chosen as the operational framework for its synergistic fit with the project’s objectives. It provided a structured yet adaptable environment to foster a user-centered platform. The Scrum methodology facilitated constant stakeholder engagement, enabling the team to incorporate feedback iteratively and ensure that the platform remains aligned with user needs.

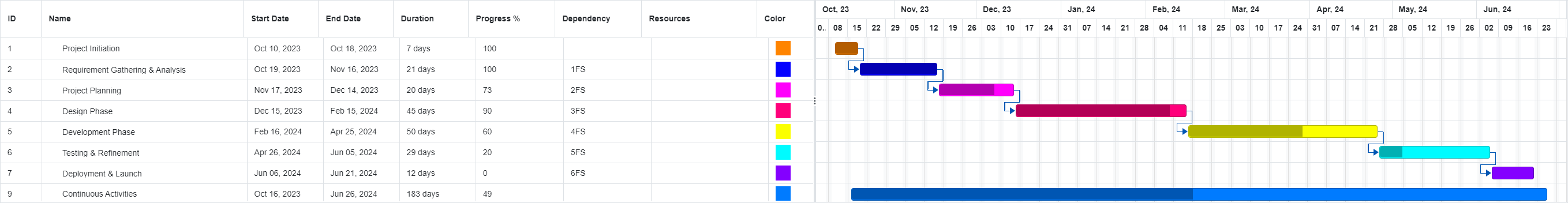
Bi-weekly sprints were scheduled, wherein the development team focused on delivering specific, prioritized features from the product backlog. Regular sprint reviews and retrospectives ensured that the project adapted to feedback and improved upon the processes continuously.

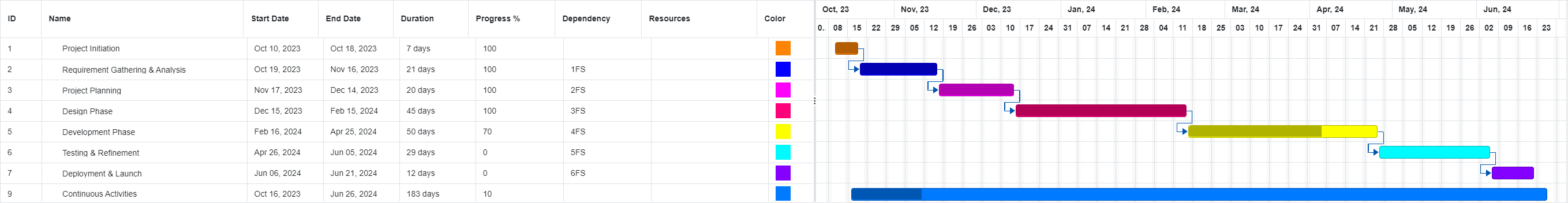
**Project Management Tools**

Trello was instrumental in organizing the workflow. It enabled the categorization of tasks into boards representing different Scrum artifacts (product backlog, sprint backlog, in-progress, and done). Trello's visual interface and ease of reorganization catered to the dynamic nature of Agile project management, providing transparency and a high-level overview of the project's status at any given time.

**Gantt Chart**

Despite Agile's emphasis on flexibility, the project's overarching timeline was charted



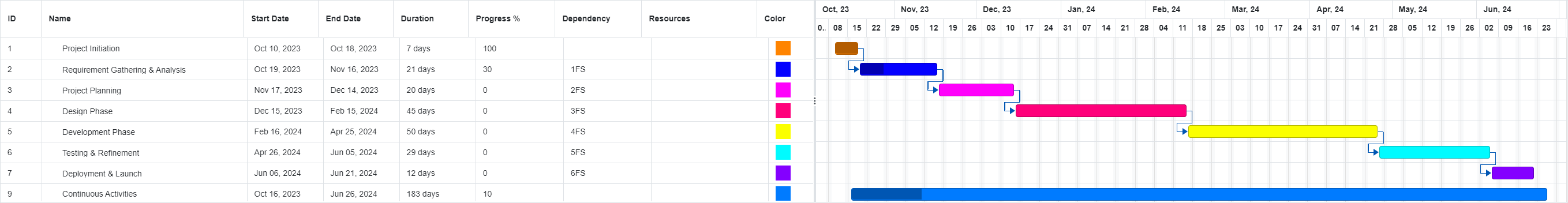
out using a Gantt chart. This served as a visual planning tool to set key milestones and task durations. It offered a macroscopic view of the project's lifecycle, ensuring adherence to the overarching deadline, and facilitated the management of dependencies between tasks.

**Version Control with GitHub**

GitHub was selected for version control, considering its robust platform for collaborative coding and its prevalence in the software development community. GitHub’s branching mechanism allowed for the isolated development of features, with

subsequent integration into the main codebase after thorough review. This practice not only ensured the integrity and continuity of the codebase but also encouraged experimental development without risking the stability of the system.

The project’s GitHub repository acted as a single source of truth for code changes, providing a comprehensive audit trail for contributions, discussions, and modifications.

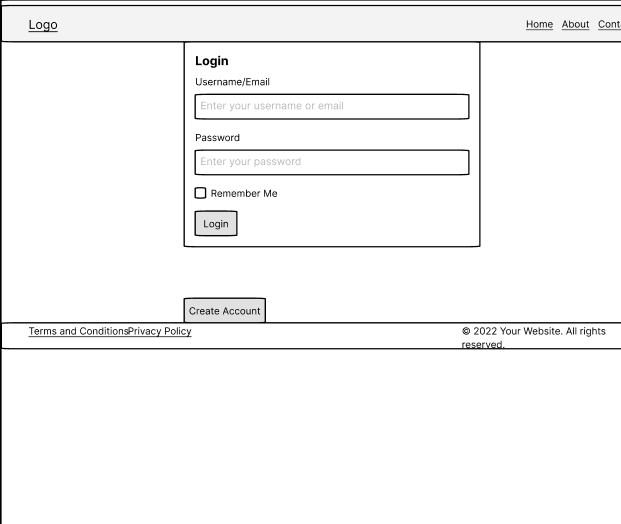


# Design Phase:

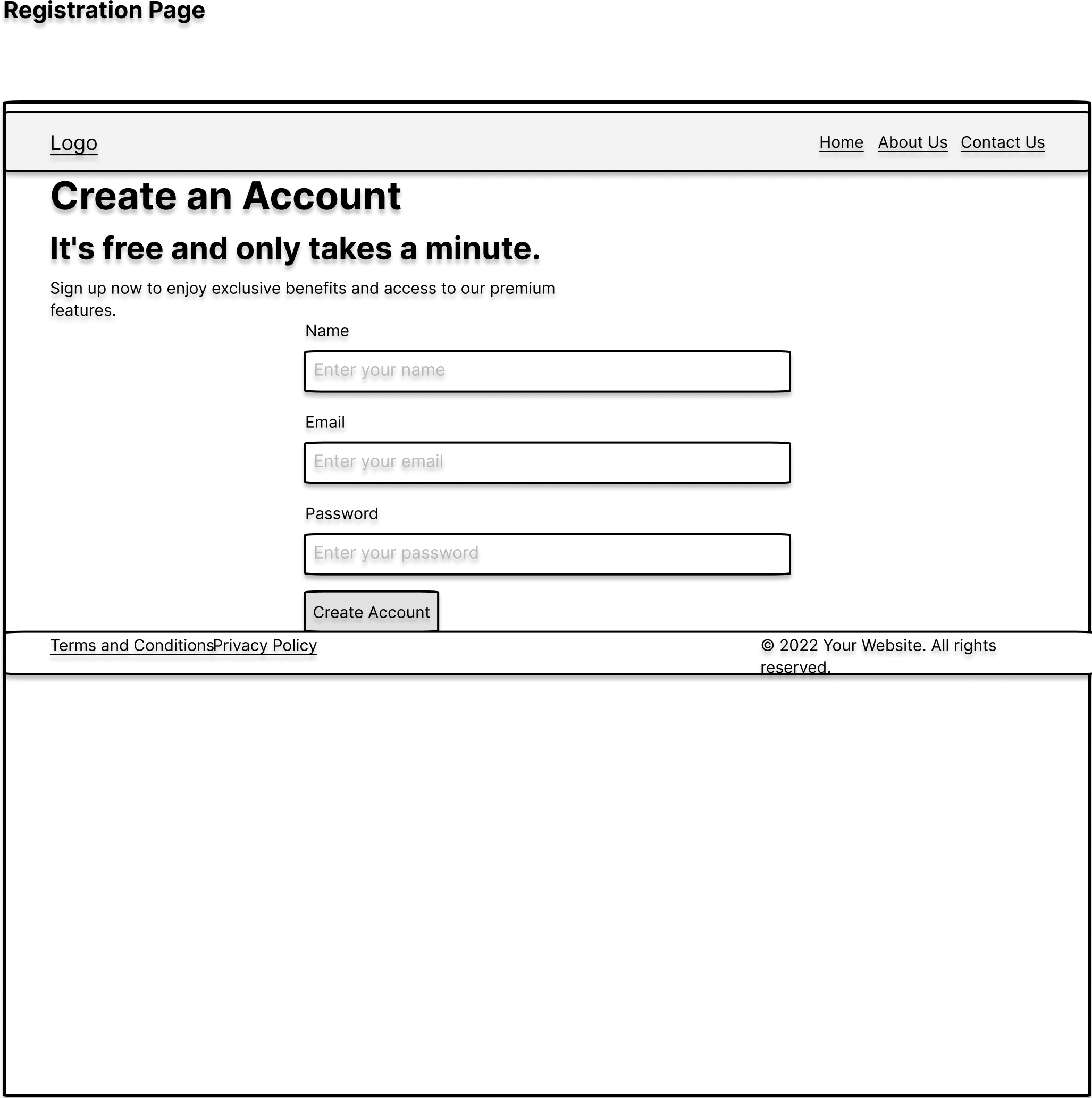
This stage marks the design phase of the application where I would be designing the required systems that would be operating within the

## Wireframes:

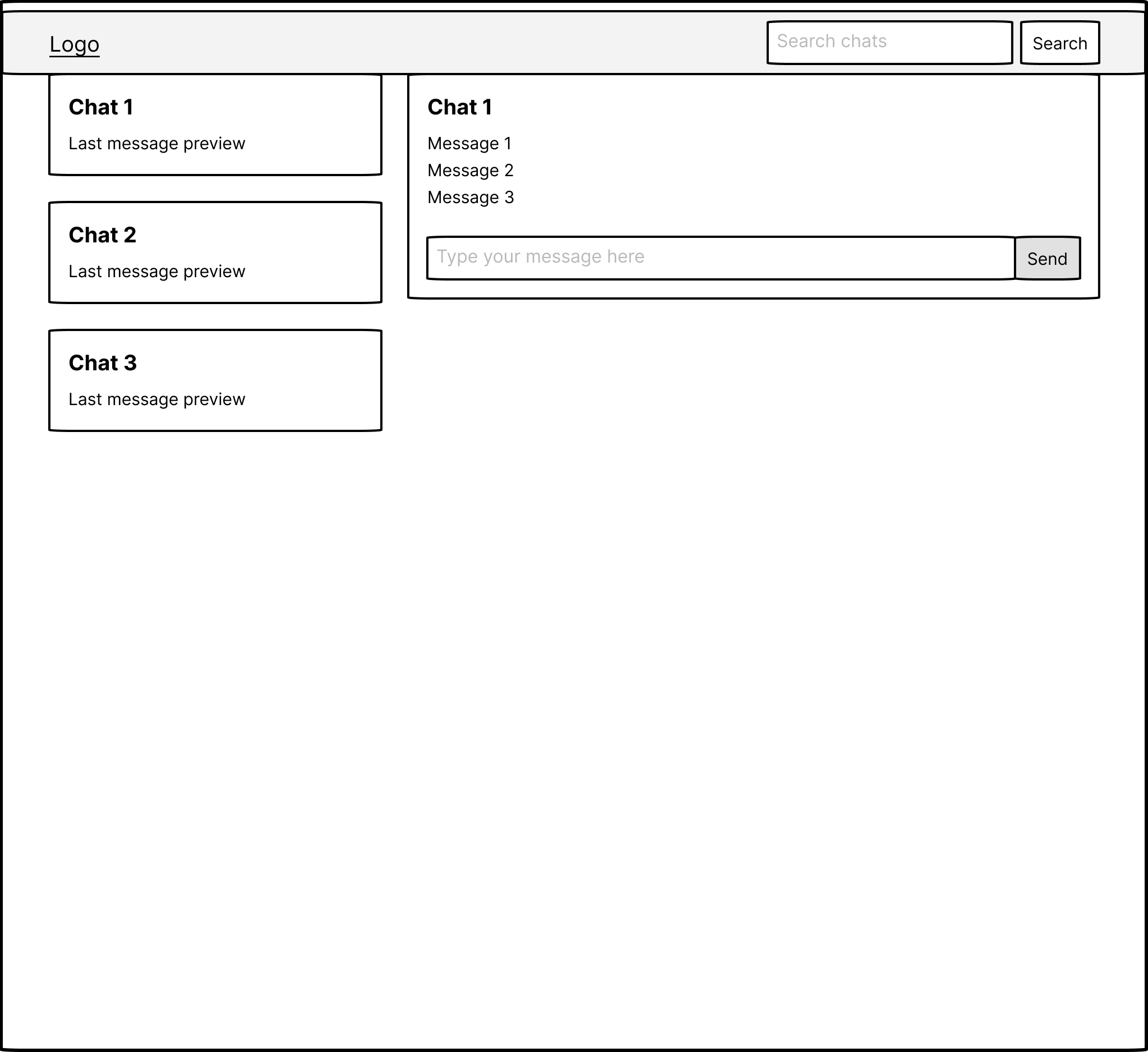
**Login:**



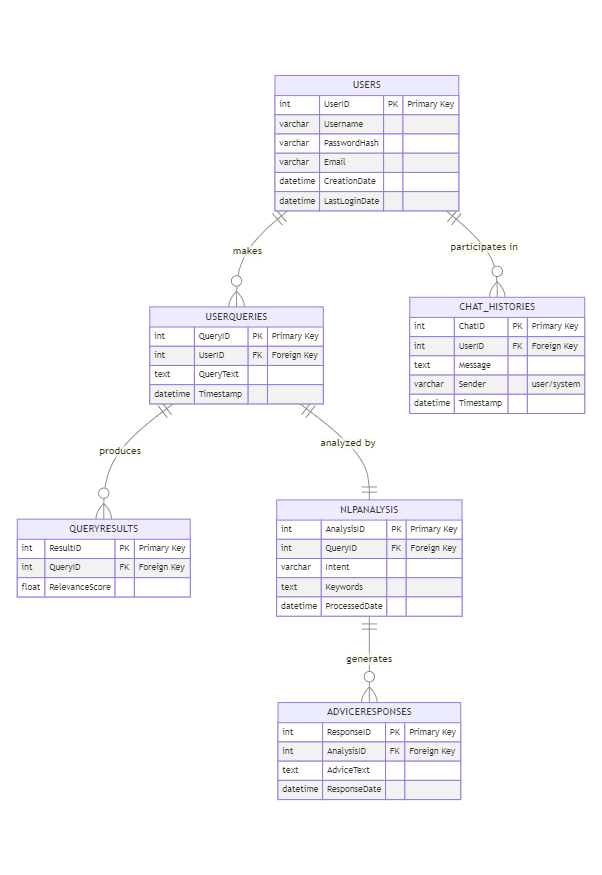
**Registration:**



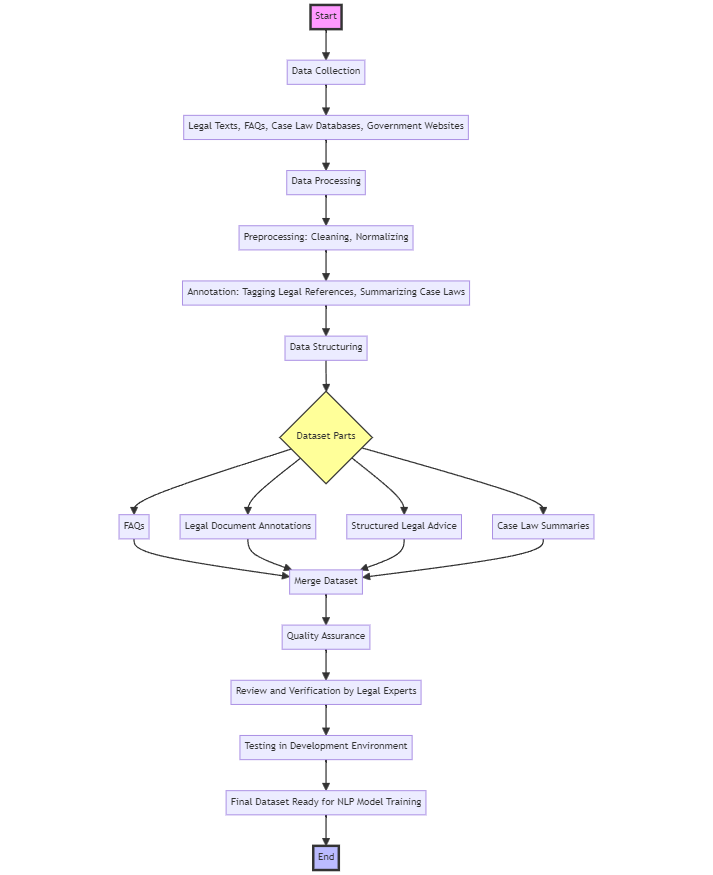
**Chat Main Page:**



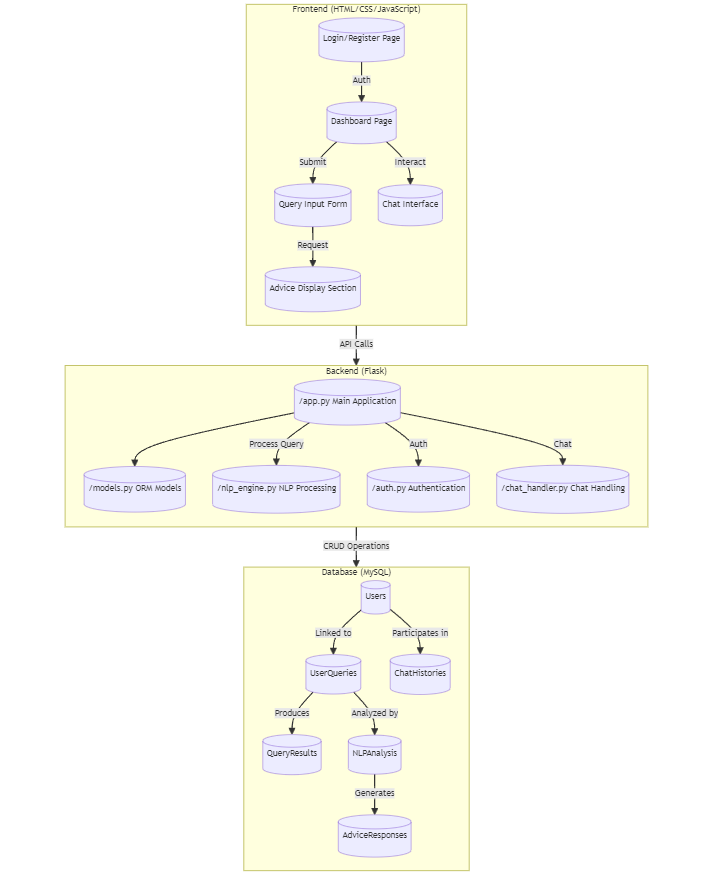
## Database Design:



## Dataset Workflow/Design:



# LLD Design:



# References

1. Khurana D, Koli A, Khatter K, Singh S. Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl. 2023;82(3):3713-3744. doi: 10.1007/s11042-022-13428-4. Epub 2022 Jul 14. PMID: 35855771; PMCID: PMC9281254.(for Image and the introduction)
2. Chomsky, N. (1965) 'Aspects of the Theory of Syntax', MIT Press, Cambridge, MA.
3. Sutskever, I., Vinyals, O., and Le, Q.V. (2014) 'Sequence to Sequence Learning with Neural Networks', in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14), MIT Press, Cambridge, MA, USA, pp. 3104–3112.
4. and Short Papers), pp. 4171–4186.
5. Hutchins, W.J. (1986) 'Machine Translation: Past, Present, Future', Ellis Horwood, Chichester, England.
6. Hutchins, W.J. (1995) 'Machine Translation: A Brief History', in Concise History of the Language Sciences: From the Sumerians to the Cognitivists, pp. 431–445.
7. Manning, C.D., and Schütze, H. (1999) 'Foundations of Statistical Natural Language Processing', MIT Press, Cambridge, MA.
8. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019) 'Language Models are Unsupervised Multitask Learners', OpenAI Blog.
9. Winograd, T. (1972) 'Understanding Natural Language', Academic Press, New York, NY.
10. Jackendoff, R. (2002) 'Foundations of Language: Brain, Meaning, Grammar, Evolution', Oxford University Press, Oxford, UK.
11. Bird, S., Klein, E., and Loper, E. (2009) 'Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit', O'Reilly Media, Inc., Sebastopol, CA.
12. Jurafsky, D., and Martin, J.H. (2019) 'Speech and Language Processing (3rd ed. draft)'
13. Hassler, M., & Fliedl, G. (2006). Text Preparation Through Extended Tokenization.(for tokensation images)
14. Jurafsky, D., and Martin, J.H. (2009) 'Speech and Language Processing (2nd ed.)', Pearson Prentice Hall, Upper Saddle River, NJ.
15. Manning, C.D. (2011) 'Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?', in Computational Linguistics and Intelligent Text Processing, Springer, Berlin, Heidelberg, pp. 171-189.
16. Smith, N.A. (2011) 'Linguistic Structure Prediction', Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers.
17. [https://www.freecodecamp.org/news/an-introduction-to-part-of-speech-tagging-and-the-hidden-markov-model-953d45338f24/](https://www.freecodecamp.org/news/an-introduction-to-part-of-speech-tagging-and-the-hidden-markov-model-953d45338f24/(linke) (link for the pos image
18. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016) 'Neural Architectures for Named Entity Recognition', in Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260–270.
19. Nadeau, D., and Sekine, S. (2007) 'A Survey of Named Entity Recognition and Classification', Linguisticae Investigationes, 30(1), pp. 3–26.
20. Chen, D., and Manning, C.D. (2014) 'A Fast and Accurate Dependency Parser using Neural Networks', in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 740–750.
21. Kübler, S., McDonald, R., and Nivre, J. (2009) 'Dependency Parsing', Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers.
22. Manning, C.D., Raghavan, P., and Schütze, H. (2008) 'Introduction to Information Retrieval', Cambridge University Press, New York, NY, USA.
23. Porter, M.F. (1980) 'An algorithm for suffix stripping', Program, 14(3), pp. 130–137.
24. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', in Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186.
25. Goodfellow, I., Bengio, Y., and Courville, A. (2016) 'Deep Learning', MIT Press, Cambridge, MA, USA.
26. Vaswani, A. et al. (2017) 'Attention is All You Need', in Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 5998–6008.
27. <https://www.codemotion.com/magazine/video/nlp-techniques-and-deep-learning/> (image for machine learning in NLP)
28. <https://sonix.ai/articles/difference-between-artificial-intelligence-machine-learning-and-natural-language-processing(image> for machine learning in NLP
29. Barocas, S., Hardt, M., and Narayanan, A. (2019) 'Fairness and Abstraction in Sociotechnical Systems', in Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 59–68.
30. Romanosky, S. (2016) 'Examining the Costs and Causes of Cyber Incidents', Journal of Cybersecurity, 2(2), pp. 121–135.
31. Voigt, P., and Von dem Bussche, A. (2017) 'The EU General Data Protection Regulation (GDPR)', Springer International Publishing, Cham.
32. Ashley, K.D. (2017) 'Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age', Cambridge University Press, Cambridge.
33. Branting, L.K. (2021) 'Explainable Artificial Intelligence and the Interpretability Problem in Legal Analytics', Jurimetrics, 61, pp. 45–68.
34. Kreutzer, R.T., and Sirrenberg, M. (2019) 'Understanding Artificial Intelligence: A Guide for Business Leaders', Springer Nature, Cham.
35. Tsarapatsanis, D., and Aletras, N. (2021) 'Machine Learning in Law: Present and Future', in Artificial Intelligence and Law, 29, pp. 355–385.
36. Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., and Sun, M. (2020) 'How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence', in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5218–5230.
37. Alarie, B., Niblett, A., and Yoon, A.H. (2018) 'How Artificial Intelligence Will Affect the Practice of Law', University of Toronto Law Journal, 68(S1), pp. 106–124.
38. Katz, D.M., Bommarito II, M.J., and Blackman, J. (2017) 'A General Approach for Predicting the Behavior of the Supreme Court of the United States', PLoS One, 12(4), e0174698.
39. Rule, C. (2017) 'Online Dispute Resolution for All: The History and Future of Online Dispute Resolution', in Online Dispute Resolution: Theory and Practice, Eleven International Publishing, The Hague, pp. 13–32.
40. Sourdin, T. (2018) 'Judge v. Robot? Artificial Intelligence and Judicial Decision-Making', University of New South Wales Law Journal, 41(4), pp. 1114–1133.