

**Course ID: CPS 5995-01**

**Software Design Document (SDD)**

**Law Digest 4 New Jersey**

**Version 1.0**

**Prepared by: Matthew Fernandez**

**Reviewed by: Joseph Tomasello and Xavier Amparo**

**Instructor: Dr. Jing-Chiou Liou**

Date: 2/14/2024

# Table of Contents

[**Table of Contents 2**](#_heading=h.bv5gdwg40ux6)

[**1.0 Introduction 3**](#_heading=h.vf5ha398yxhl)

[1.1 Goals and Objectives 3](#_heading=h.dak9r6ebkl35)

[1.2 Statement of Scope 3](#_heading=h.gfbxltyz4rw8)

[1.3 Software Context 4](#_heading=h.vw10jo8r7qg)

[1.4 Major Constraints 5](#_heading=h.rlrde31po41k)

[**2.0 Data Design 5**](#_heading=h.qkfqhmp51xqt)

[2.1 Internal Software Data Structure 5](#_heading=h.tcdsvq3z0erb)

[2.2 Global Data Structure 5](#_heading=h.v77n288e2mpg)

[2.3 Temporary Data Structure 6](#_heading=h.7095j54e3x52)

[2.4 Database Description 6](#_heading=h.pzfn09b7nu1u)

[**3.0 Architectural and Component-Level Design 7**](#_heading=h.oo2a5xuq8kqk)

[3.1 System and Program Structure 7](#_heading=h.pv53098vg00f)

[3.2 Architecture diagram showing system components 8](#_heading=h.bwys8vizggzh)

[3.3 Interface descriptions 8](#_heading=h.ep7e05s091ge)

[3.4 External machine interfaces 8](#_heading=h.ueadkd81k6e)

[3.5 Human interface 9](#_heading=h.3a2w8hgckncq)

[**4.0 User interface design 9**](#_heading=h.8xf6hwaxegu7)

[4.1 Description of the user interface 9](#_heading=h.pmknkp9jbf3c)

[4.2 Screen images 11](#_heading=h.z3iab2k5cfs9)

[4.3 Objects and actions 11](#_heading=h.tnd2529rhpap)

[4.3.1 Landing Page 11](#_heading=h.26ebealrug9m)

[4.3.2 Sign-Up Page 12](#_heading=h.1obei97vykuu)

[5.1 Project Constraints 12](#_heading=h.35nkun2)

[5.2 Critical Project Barriers 12](#_heading=h.1ksv4uv)

[**6.0 Testing issues 12**](#_heading=h.cy0ssnwr7acv)

[6.1 Testing of data requirements 12](#_heading=h.n4ssrr7n3bwc)

[6.2 Testing of functionality 12](#_heading=h.b94voupaj3gf)

[6.3 Testing of component interfaces 13](#_heading=h.st0dqepm1ksl)

[6.4 Testing of GUI’s 13](#_heading=h.rpe87y83xwuf)

# 

# 1.0 Introduction

This document is for the purposes of describing the components and functions making up the Legal Digest 4 New Jersey system. This document illustrates the system architecture, use cases, and other requisite information for the purpose of informing readers the system specifications on a low level.

## 1.1 Goals and Objectives

The objective of this system design document is to provide an initial description of the Law Digest 4 New Jersey system to inform the development process. Our system must be responsive, quick to amend with changes in the law, and must interact with both user prompted searches of our data repository and of our AI language model. Since this project follows an agile-based development methodology, alterations and updates may be incorporated based on client feedback.

## 1.2 Statement of Scope

This Software Design Document will serve as a baseline for building the Law Digest 4 New Jersey system and is focused on the core, critical components.

The following is a list of diagrams that can be expected to be contained within this document:

* System Architecture
* Context Diagram
* Representation of functional modules
* Database schema Diagram

## 1.3 Software Context

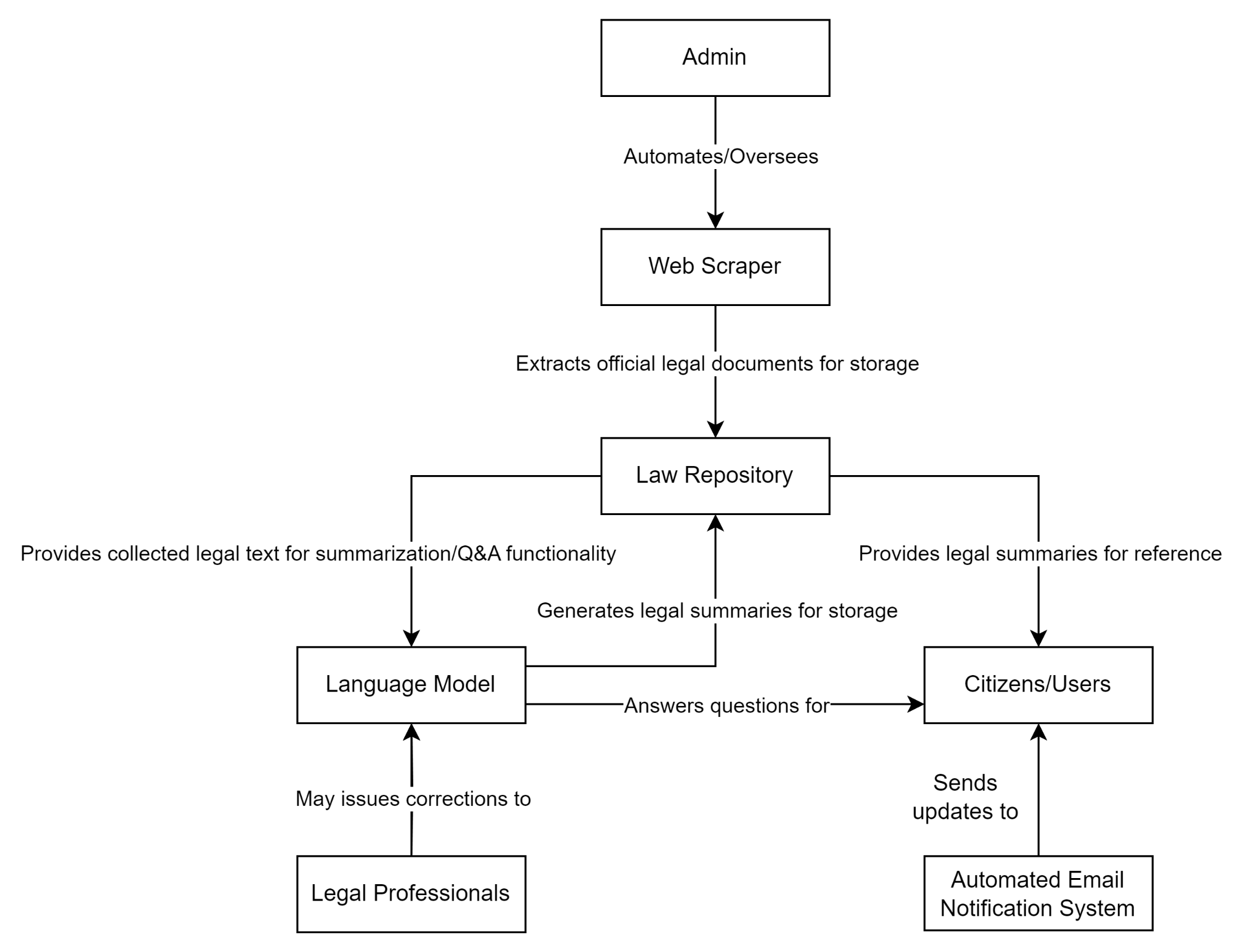


Figure 1 - Context Diagram for LD4NJ

This diagram represents the entities which interact with the Law Digest 4 New Jersey system.

Admins: Admins/developers oversees the web scraper that supplies both the repository with legal documents and the Large Language Model with training data.

Language Model: Language model is trained to both summarize the documents it has been trained on, as well as to answer questions prompted by the user.

Law Repository: Repository exists to provide users with the source documents upon which our site is designed to summarize and take questions on. This gives the users transparency and the knowledge that the information provided by our LLM is genuine.

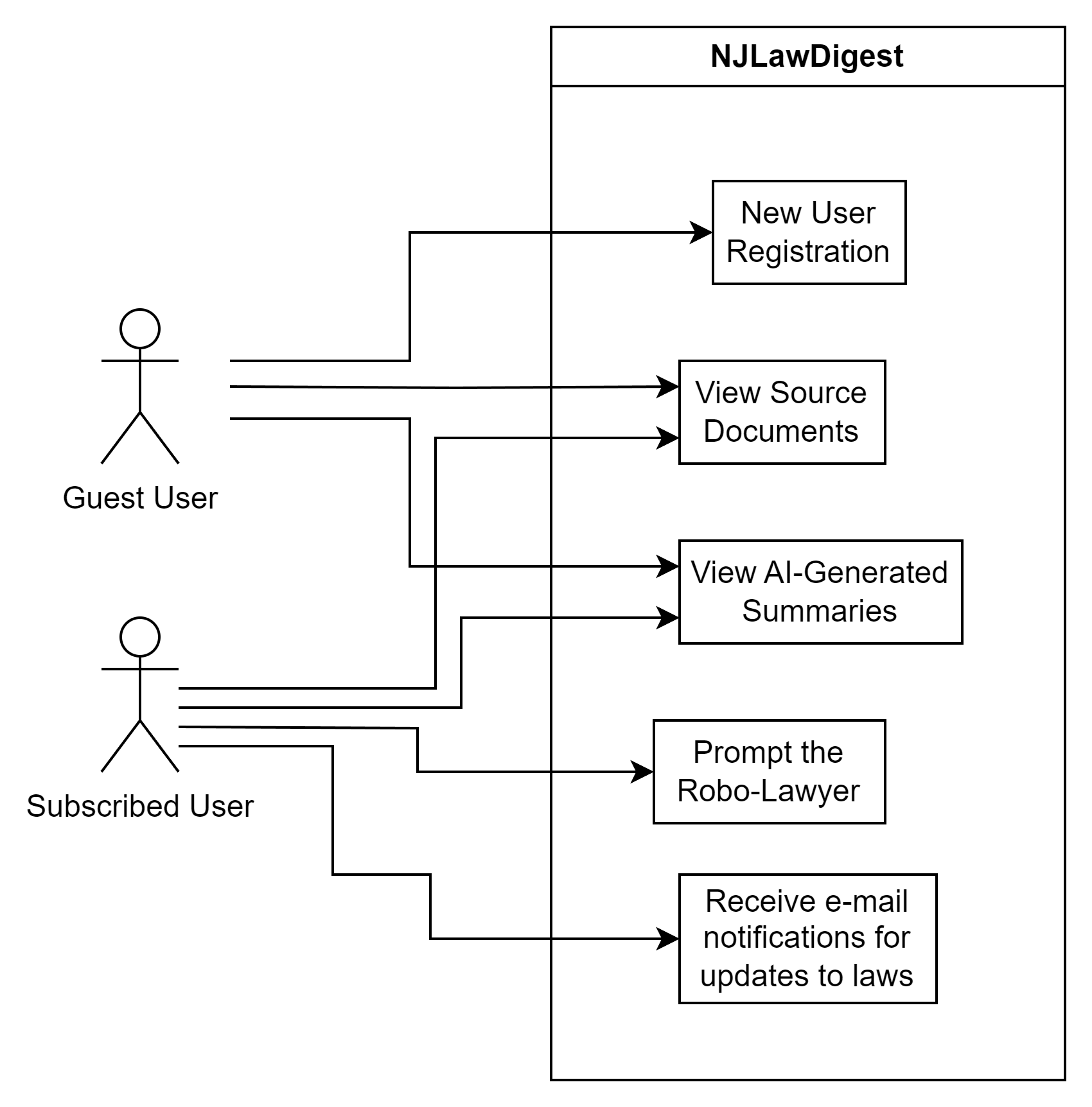
Legal Professionals: As the domain experts, lawyers and other such professionals may be necessary to audit our system to eliminate inaccuracies or outdated information.

Citizens/Users: Our end user is a New Jersey citizen, in need of pertinent information and updates regarding state ordnance and regulations.

## 1.4 Major Constraints

Subject matter (legal) experts may be required to verify our information.

**1.5 Use Cases**

****

# 2.0 Data Design

## 2.1 Internal Software Data Structure

The data that is passed from the web scraper to the repository and the LLM will come as text documents, either as .doc or .docx, or txt files depending on the source of the information. This data, once analyzed and summarized, is made available as plain text, with the user able to view the source document in addition to the AI-generated summary. Legal professionals may correct our LLM, with which a note will pop up on any relevant bills/documents that contain the input from such professionals.

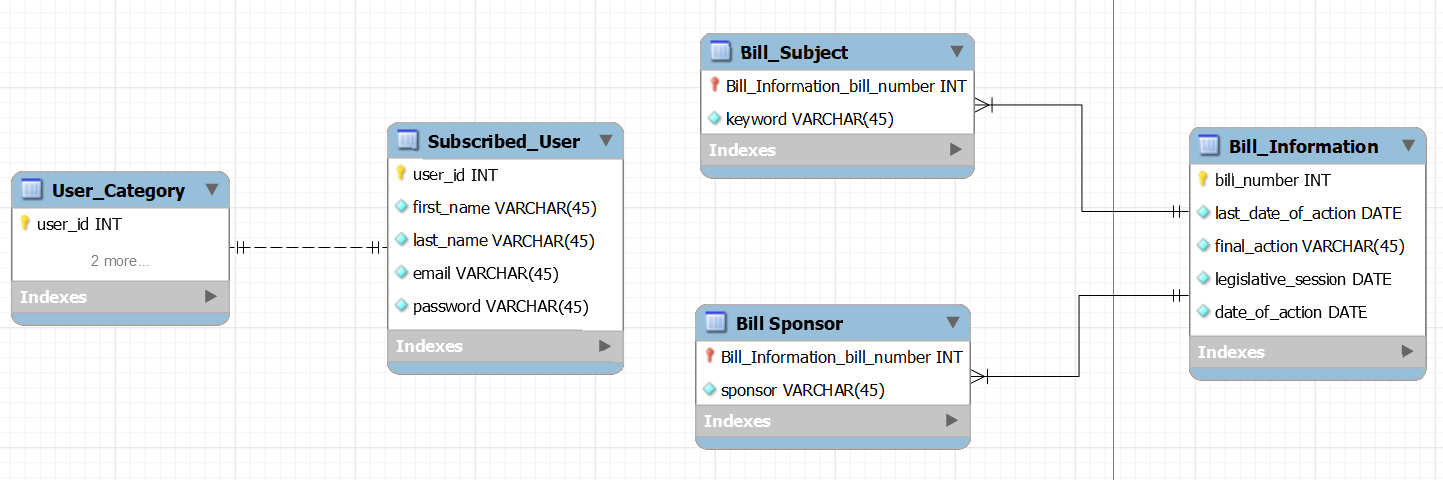
## 2.2 Global Data Structure

The only data that will remain unchanged throughout our system are source documents taken from existing state-managed resources.

## 2.3 Temporary Data Structure

The tensors of the LLM will only be relevant for the training and evaluation phases, although continuous training and evaluation will be necessary to keep up-to-date with new laws.

## 2.4 Database Description

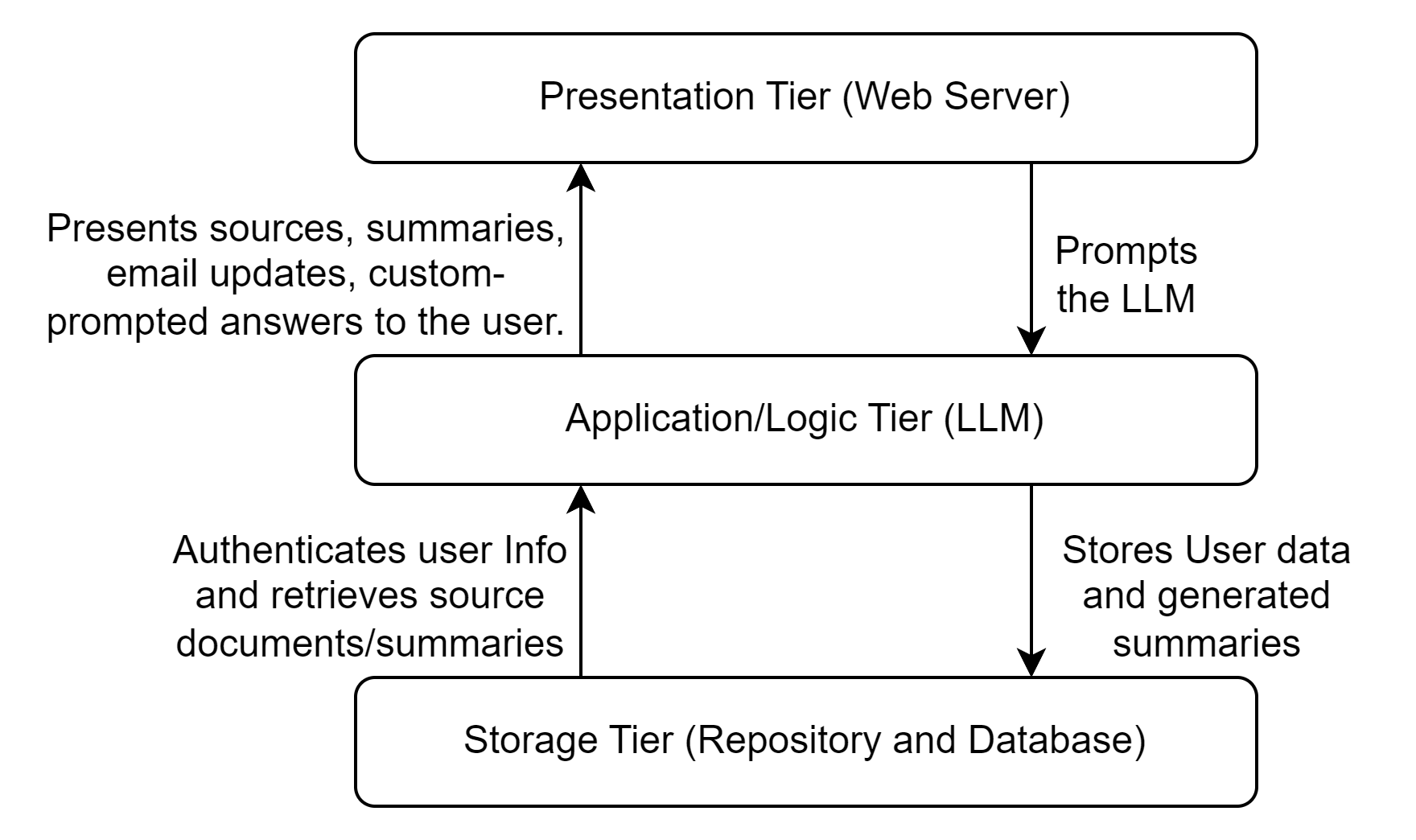
****

This schema was drafted in MySQL Workbench but will be written in Supabase.

* The User\_Category denotes if the user is subscribed
* Subscribed\_User stores the personal information of the user
* Bill\_Information contains the information on the bill itself
  + Bill\_Sponsor has a N:1 relationship with Bill\_Information as one bill can have multiple sponsors.
  + In a similar fashion, Bill\_Subject has a N:1 relationship with Bill\_Information as 1 bill can be in multiple categories.

# 3.0 Architectural and Component-Level Design

## 3.1 System and Program Structure

****

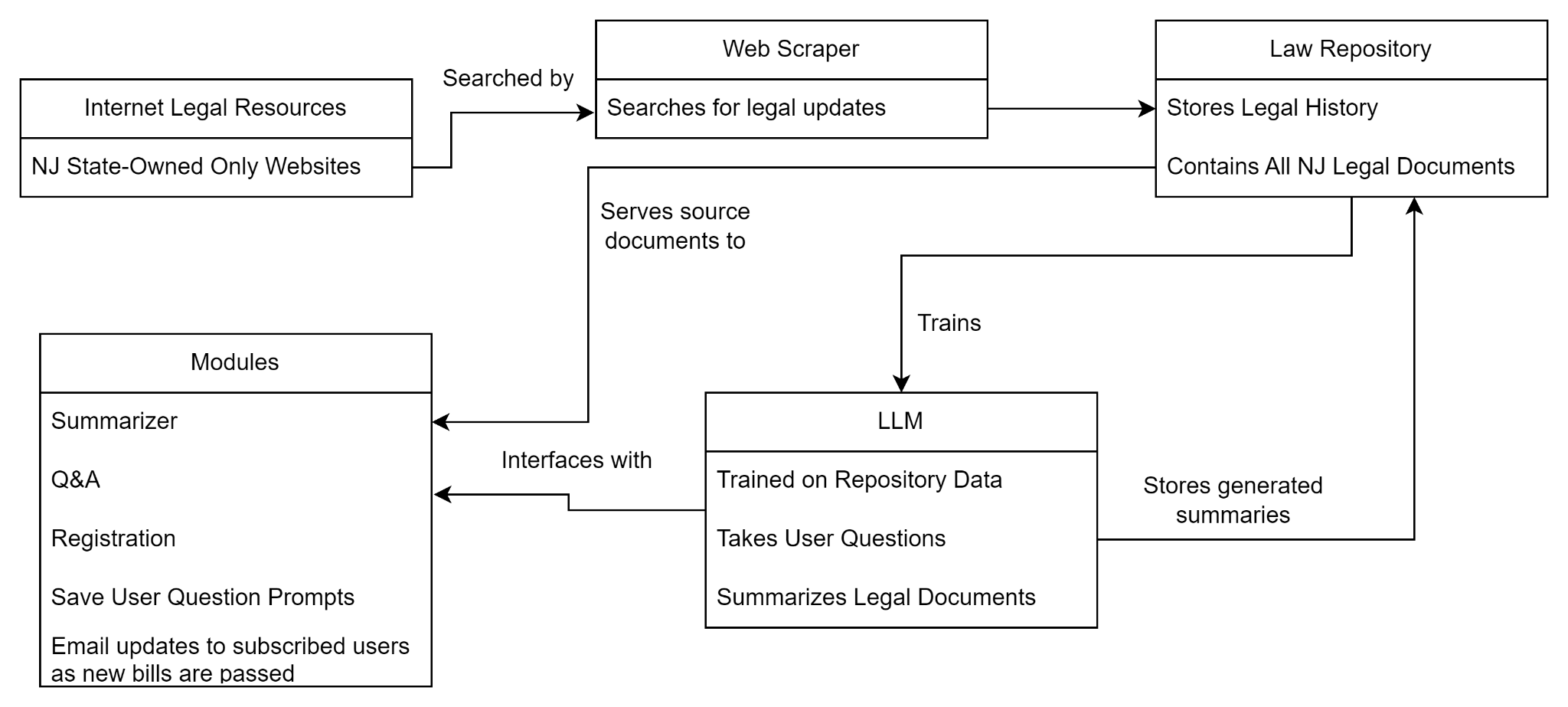
Presentation Tier: HTML, PHP, and CSS. Our site will be developed and tested in a Chromium environment. We will not be testing on Firefox, Microsoft Edge, etc. The Web Server will be hosted locally on XAMPP for development and testing.

Application Tier: The LLM will have to be hosted on a computer with a powerful enough GPU to perform the necessary computations for interaction. It will then be accessed via PHP from the Presentation Layer.

Repository Tier: A Supabase database will be used to store users and legal document information.

A device with a machine-learning capable GPU will be used to host all these computationally expensive services (LLM, Repository, Web Scraper) during development.

## 3.2 Architecture diagram showing system components

****

## 3.3 Interface descriptions

1. Website - Main function is search through legal source documents and their AI-generated summaries. English language support only.
   1. Input: Language model, documents from the web scraper, user database.
   2. Output: Up-to-date legal documentation, registered user, responsive AI.
2. Language model (Robot Lawyer) - Integrated with the website to answer questions not answered by the summary of laws. Users must agree that our system will not be held liable for any actions suggested by the AI before being allowed to use it.
   1. Input: Text-based training data, user-prompted questions
   2. Output: Relevant answers pertaining to user questions and backed by knowledge acquired from legal documentation.
3. Web Scraper - On the back end, provides the LLM with new training data, and uploads new documents to our website.
   1. Input: NJ-sponsored and managed legal resources
   2. Output: Plaintext, unformatted documents to feed to the LLM and to the front end website.

## 3.4 External machine interfaces

3.4.1 In our system’s lifetime, it is possible we may require more hardware to host our service for all our users 24/7. As new laws are passed and added to the official website of the New Jersey Office of Legislative Services, our dataset of legal documents and summaries will likewise need to grow to accommodate them. Depending on the rate of growth, our database may have to be migrated to its own server, separate from any LLM, web scraping, or website resources in order to have enough space, while still being able to interact with our system’s other components.

3.4.2 Bootstrap will be integrated into our front end CSS via a hyperlink, meaning that constant access between the client and internet is necessary.

3.4.3 The user, when accessing our system and its services, should be notified that the legal information available through the website is updated on a weekly basis, not in real time.

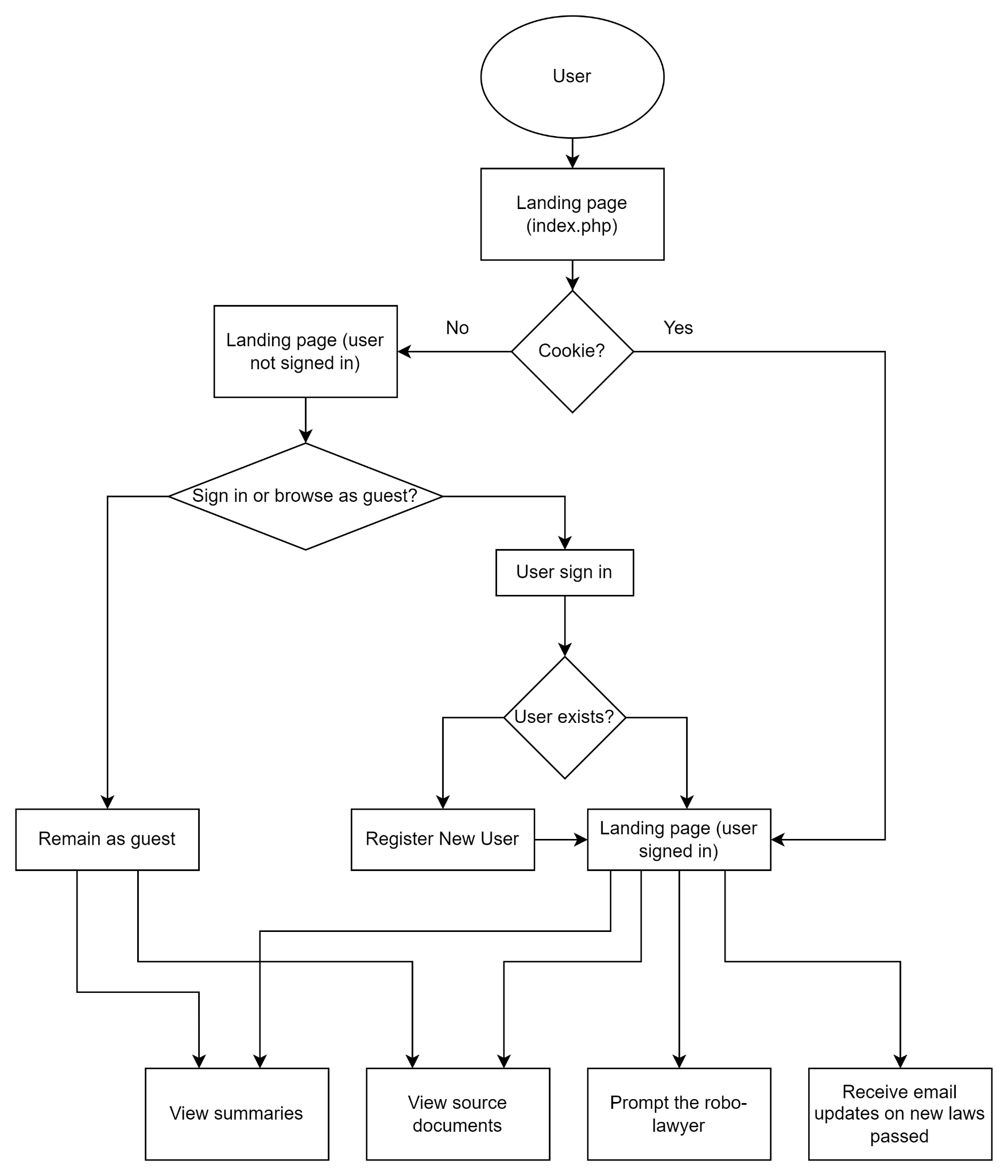
## 3.5 Human interface

The website has 2 main pages, a main landing page for guest and subscribed users, and a registration page for users who wish to submit their email to us so they can use our “robot lawyer”.

# 4.0 User interface design

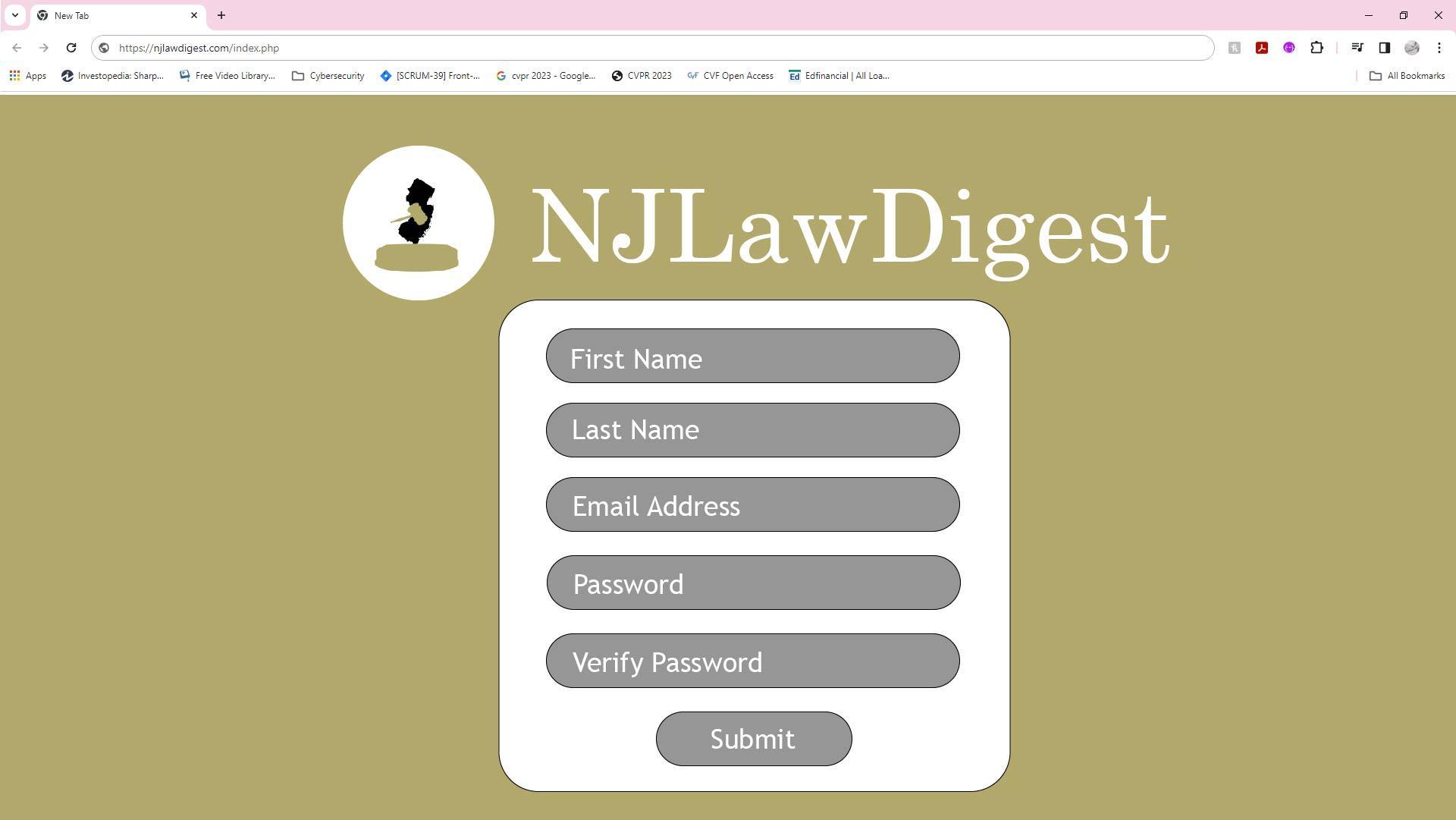
## 4.1 Description of the user interface

Username is displayed on top, the user can search by category in the top search bar. Search results will appear in the bottom right, and the AI interface always stays to the left.



## 4.2 Screen images



****

## 4.3 Objects and actions

### 4.3.1 Landing Page

* Sign Out function
* Search function
* AI prompt function (For Registered User)
* Document and Search Result viewing window

### 4.3.2 Sign-Up Page

* User registration takes first and last names, email address, and a password.
* Submit button only creates a new user; no two-factor authentication email is sent.

**5.0 Restrictions, limitations and constraints**

## 5.1 Project Constraints

The following represent known project constraints:

* Subject matter (legal) experts are required to verify our information.

## 5.2 Critical Project Barriers

Unlike risks, critical project barriers are intractable issues that can be critical to a project’s initiative and in our case, consist of the following:

* Exponential expansion of dataset as new bills are created and collected
* Dependency on the official website of the New Jersey Office of Legislative Services for the collection of legislative data

# 6.0 Testing issues

## 6.1 Testing of data requirements

* Legal documents on our site must be refreshed once a week to be kept up-to-date

## 6.2 Testing of functionality

* Guest Users: Search legal information, apply search filters (topic and session), view summaries, register for notifications
* Registered/Subscribed Users: All guest user tests, Email notifications, LLM access.

## 6.3 Testing of component interfaces

* Data from web scraper must be un-formatted correctly (i.e. page numbers and formatting must not affect training of LLM)
* LLM must provide summaries that we (non-legal professionals) can understand.

## 6.4 Testing of GUI’s

* Legibility of text
* CSS errors leading to improper loading of HTML elements