**. User Uploads a Document (PDF, DOCX, TXT)**

**File Involved:**

* resources/views/document/create.blade.php (upload form)
* app/Http/Controllers/DocumentController.php (store method)

**Workflow:**

* User uploads a document via the web form.
* The file is sent to the store() method of DocumentController.

**2. PHP Extracts Text from the Uploaded File**

**File Involved:**

* app/Http/Controllers/DocumentController.php

**Workflow:**

* PHP checks the file extension:
* If .txt: Reads the text directly.
* If .pdf: Uses smalot/pdfparser to extract text.
* If .docx: Uses phpoffice/phpword to extract text.
* The extracted text is stored in a variable ($documentText).

**3. PHP Sends the Text to the SpaCy API (Python)**

**Files Involved:**

* app/Http/Controllers/DocumentController.php
* spacy\_api.py (your Python Flask API)

**Workflow:**

* PHP sends a POST request to http://127.0.0.1:5050/classify with the extracted text.
* Example payload: { "text": "This is the content of the document..." }

**4. SpaCy API Processes the Text**

**File Involved:**

* spacy\_api.py

**Workflow:**

* SpaCy analyzes the text:
* Extracts **named entities** (e.g., people, dates, organizations).
* Classifies the document (e.g., Contract, NDA, Policy) using rules or ML.
* SpaCy returns a JSON response:

json

Apply to index.blade....

  {

    "entities": [

      {"text": "Juan", "label": "PERSON"},

      {"text": "July 1", "label": "DATE"}

    ],

    "category": "Contract"

  }

**5. PHP Saves the Category to the Database**

**Files Involved:**

* app/Http/Controllers/DocumentController.php
* app/Models/Document.php
* database/migrations/2025\_07\_16\_161050\_add\_category\_to\_documents\_table.php (migration for category column)

**Workflow:**

* The category from SpaCy is saved to the category column of the documents table.
* The document record is created in the database.

**6. PHP Passes Entities and Category to the Blade View**

**Files Involved:**

* app/Http/Controllers/DocumentController.php (show method)
* resources/views/document/show.blade.php

**Workflow:**

* When viewing a document, PHP again extracts the text, sends it to SpaCy, and gets the entities.
* The controller passes both the document and the entities to the Blade view.

**7. Blade View Displays the Results**

**Files Involved:**

* resources/views/document/show.blade.php

**Workflow:**

* The view displays:
* The document’s **category** (e.g., Contract, NDA, etc.)
* The **named entities** as a list (e.g., Juan (PERSON), July 1 (DATE))

**8. LegalController Integration**

**Files Involved:**

* app/Http/Controllers/LegalController.php
* resources/views/legal/show.blade.php (if you display entities here)

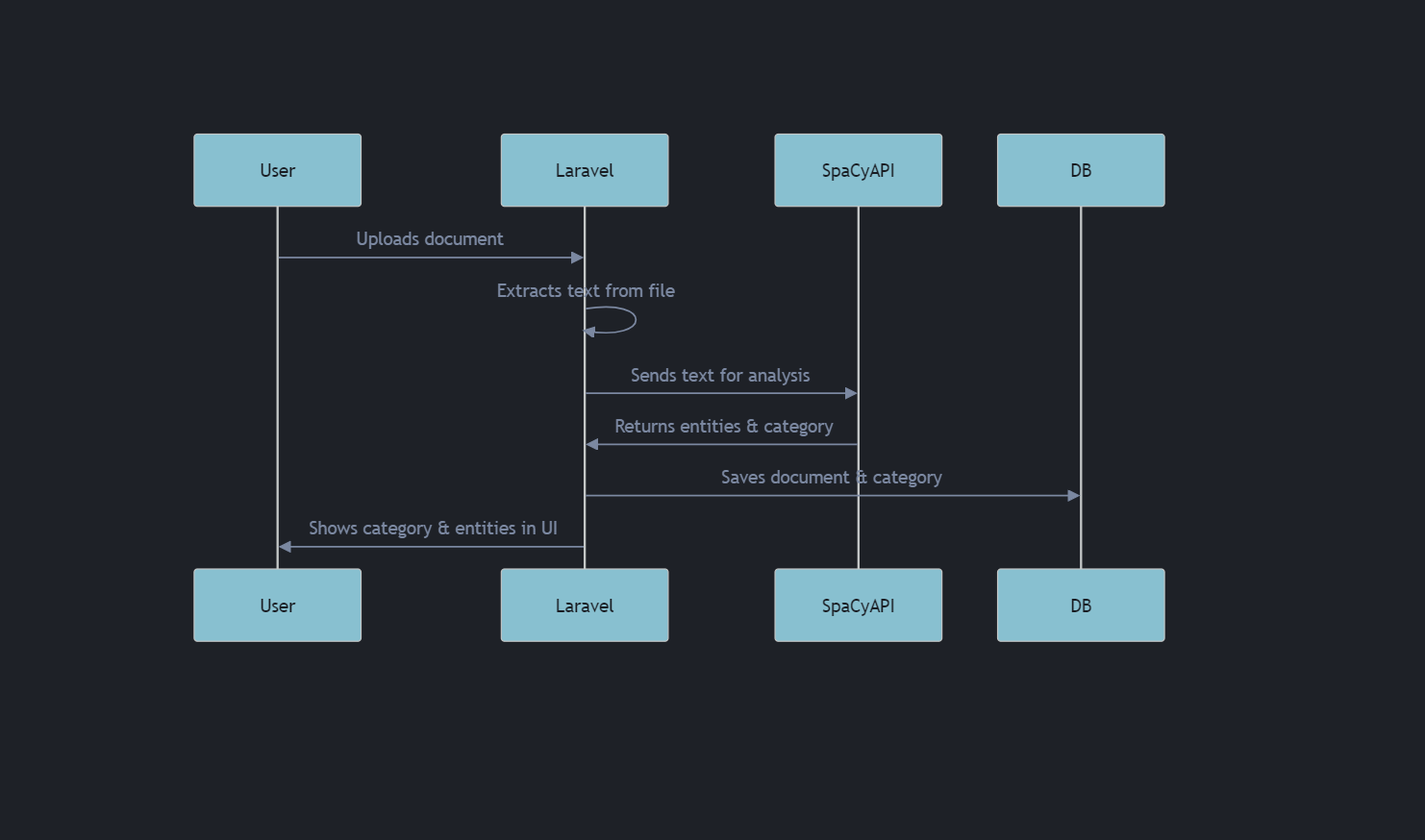
**Workflow:**

* When a document is approved in the legal module, the same text extraction and SpaCy classification is performed.
* The document’s category is updated.
* Entities can also be displayed in the legal view.

**Summary Table**

| **Step** | **What Happens** | **Main Files Involved** |
| --- | --- | --- |
| 1 | User uploads document | create.blade.php, DocumentController.php |
| 2 | PHP extracts text | DocumentController.php |
| 3 | PHP sends text to SpaCy | DocumentController.php, spacy\_api.py |
| 4 | SpaCy analyzes text | spacy\_api.py |
| 5 | PHP saves category | DocumentController.php, Document.php, migration file |
| 6 | PHP passes data to view | DocumentController.php, show.blade.php |
| 7 | Blade view displays results | show.blade.php |
| 8 | LegalController integration | LegalController.php, legal/show.blade.php |

**Visual Workflow Diagram**

****