

"""

Input: person knows a person has a will and a trust or some legal documents

Context: Grab info from the user, name of the person, name of the person's lawyer,

Another agent will up the form,

Output: Create a PDF file with all of the information with the same headings

"""

from typing import Optional

from pydantic import BaseModel, Field

from swarm\_models.openai\_function\_caller import OpenAIFunctionCaller

PROBABE\_SYS\_PROMPT = """

### System Prompt: Autonomously Fill Out Probate Form DE-122/GC-322

---

**\*\*Task Description:\*\***

You are an intelligent LLM agent tasked with filling out the Probate Form DE-122/GC-322 based on the users profile information. The form includes fields such as attorney details, court information, case number, and more. You must extract relevant information from the user's profile, fill out the form fields accurately, and mark the appropriate checkboxes.

---

### ### Step-by-Step Instructions:

#### 1. **\*\*Initialize Profile Data:\*\***

- Begin by loading the users profile data. This profile may include details like the attorney's name, state bar number, address, court details, case information, and any specific instructions provided by the user.

#### 2. **\*\*Form Field Mapping:\*\***

- Map the user's profile data to the corresponding form fields. Use the Pydantic schema provided to ensure that each form field is filled with the appropriate data.

- The schema fields include: `attorney\_name`, `state\_bar\_number`, `court\_name`, `case\_number`, `decedent\_name`, and so on.

#### 3. **\*\*Field Filling:\*\***

- For each field in the form:
  - **\*\*Attorney Information:\*\***
    - Populate `attorney\_name` with the user's attorney name.
    - Populate `state\_bar\_number` with the user's state bar number.
    - Fill in `attorney\_address`, `court\_name`, `case\_number`, etc., with the corresponding data from the profile.
  - **\*\*Court Information:\*\***
    - Fill out `court\_name`, `court\_address`, `hearing\_date`, `hearing\_time`, `hearing\_dept`, and `hearing\_room` based on the provided court details.
  - **\*\*Decedent Information:\*\***

- Populate the `decedent\_name` field with the name of the decedent or trust involved in the probate.

#### 4. **Checkbox Selection:**

- Evaluate the circumstances based on the users profile:
  - Select `as\_individual`, `as\_person\_cited`, and other relevant checkboxes depending on the users role and the type of service.
  - For example, if the user is serving as an individual, mark `as\_individual: True`.
  - Review legal codes and guidelines associated with each checkbox (e.g., `Code Civ. Proc., § 416.10`) and ensure that the correct checkboxes are selected based on the profile data and case type.

#### 5. **Service Details:**

- If the user profile includes details about how the citation was served (e.g., `served\_by\_personal\_delivery`), fill out the relevant checkboxes and fields (`service\_person\_name`, `service\_date`, `service\_time`).
  - Include additional information under `service\_other\_details` if specific instructions are provided.

#### 6. **Validation and Final Review:**

- After filling out all fields and checkboxes, review the completed form to ensure accuracy and completeness.
- Cross-check the filled data with the users profile to confirm that all relevant information has been included.

#### 7. **Finalize and Prepare for Submission:**

- Mark the `acknowledgement\_checkbox` if the user has acknowledged the information.

- Insert any final signatures, dates, or additional required fields (`declarant\_signature`, `declarant\_date`).
- Save the completed form for review by the user or automatically submit it based on the user's instructions.

8. **\*\*Provide Output:\*\***

- Return the filled form as a structured data object or PDF, ready for printing or digital submission.
- Optionally, provide a summary of the filled form fields and selected checkboxes for the user's review.

---

**\*\*Example Input:\*\***

- User profile includes attorney name: "John Doe", state bar number: "123456", court name: "Superior Court of California", decedent name: "Jane Smith", etc.

**\*\*Example Output:\*\***

- The Probate Form DE-122/GC-322 filled with the above data, with all relevant checkboxes and fields correctly populated.

""

```
class CitationForm(BaseModel):
```

```
    attorney_name: Optional[str] = Field(None, alias="FillText7")
```

```
    state_bar_number: Optional[str] = Field(None, alias="FillText9")
```

attorney\_address: Optional[str] = Field(None, alias="FillText10")

court\_name: Optional[str] = Field(None, alias="FillText11")

decedent\_name: Optional[str] = Field(None, alias="FillText12")

case\_number: Optional[str] = Field(None, alias="FillText13")

hearing\_date: Optional[str] = Field(None, alias="FillText14")

hearing\_time: Optional[str] = Field(None, alias="FillText15")

hearing\_dept: Optional[str] = Field(None, alias="FillText16")

hearing\_room: Optional[str] = Field(None, alias="FillText17")

court\_address: Optional[str] = Field(None, alias="FillText18")

clerk\_name: Optional[str] = Field(None, alias="FillText19")

deputy\_name: Optional[str] = Field(None, alias="FillText51")

service\_person\_name: Optional[str] = Field(

None, alias="FillText54"

)

service\_person\_address: Optional[str] = Field(

None, alias="FillText1"

)

service\_person\_telephone: Optional[str] = Field(

None, alias="FillText2"

)

service\_date: Optional[str] = Field(None, alias="FillText3")

service\_time: Optional[str] = Field(None, alias="FillText56")

# Checkboxes for different roles or actions

as\_individual: Optional[bool] = Field(False, alias="CheckBox9")

as\_person\_cited: Optional[bool] = Field(False, alias="CheckBox8")

```
under_code_civ_proc_416_10: Optional[bool] = Field(
    False, alias="CheckBox7"
)
```

```
under_code_civ_proc_416_20: Optional[bool] = Field(
    False, alias="CheckBox6"
)
```

```
under_code_civ_proc_416_40: Optional[bool] = Field(
    False, alias="CheckBox5"
)
```

```
under_code_civ_proc_416_60: Optional[bool] = Field(
    False, alias="CheckBox4"
)
```

```
under_code_civ_proc_416_90: Optional[bool] = Field(
    False, alias="CheckBox3"
)
```

# Further information for detailed service

```
served_by_personal_delivery: Optional[bool] = Field(
    False, alias="CheckBox2"
)
```

```
served_by_substituted_service: Optional[bool] = Field(
    False, alias="CheckBox1"
)
```

```
service_other_details: Optional[str] = Field(
    None, alias="FillText57"
)
```

# Additional fields for various types of services

registered\_process\_server: Optional[bool] = Field(

False, alias="CheckBox55"

)

exempt\_from\_registration: Optional[bool] = Field(

False, alias="CheckBox56"

)

other\_service\_type: Optional[str] = Field(

None, alias="FillText85"

)

service\_fees: Optional[str] = Field(None, alias="FillText105")

registration\_no: Optional[str] = Field(None, alias="FillText104")

county: Optional[str] = Field(None, alias="FillText103")

expiration\_date: Optional[str] = Field(None, alias="FillText102")

# Additional checkboxes and fields based on service details

other\_details\_1: Optional[str] = Field(None, alias="FillText101")

other\_details\_2: Optional[str] = Field(None, alias="FillText100")

# Other options and related fields

related\_case\_name: Optional[str] = Field(None, alias="FillText97")

related\_case\_checkbox: Optional[bool] = Field(

False, alias="CheckBox78"

)

related\_case\_info: Optional[str] = Field(None, alias="FillText88")

# Fields for declarations and signature

declarant\_signature: Optional[str] = Field(

    None, alias="FillText91"

)

declarant\_date: Optional[str] = Field(None, alias="FillText90")

declarant\_location: Optional[str] = Field(

    None, alias="FillText89"

)

# Final section for confirmation and acknowledgments

acknowledgement\_checkbox: Optional[bool] = Field(

    False, alias="CheckBox80"

)

acknowledgement\_date: Optional[str] = Field(

    None, alias="FillText87"

)

acknowledgement\_signatory: Optional[str] = Field(

    None, alias="FillText53"

)

acknowledgement\_title: Optional[str] = Field(

    None, alias="FillText52"

)

# Footer notices

footer\_notice\_1: Optional[str] = Field(



```

    None, alias="NoticeHeader1"

)

footer_notice_2: Optional[str] = Field(

    None, alias="NoticeFooter1"

)


# Reset and submit actions (for informational purposes)

reset_form: Optional[bool] = Field(False, alias="ResetForm")

save_form: Optional[bool] = Field(False, alias="Save")

print_form: Optional[bool] = Field(False, alias="Print")


# Example usage:

# Initialize the function caller

model = OpenAIFunctionCaller(

    system_prompt=PROBABLE_SYS_PROMPT,

    max_tokens=3500,

    temperature=0.9,

    base_model=CitationForm,

    parallel_tool_calls=False,

)

out = model.run(

    "Create a game in python that has never been created before. Create a new form of gaming

experience that has never been contemplated before."

)

```

print(out)