

UML Assistant



OVERVIEW

The **LLM Generator Assistant** is an AI-powered tool that helps you create UML diagrams by simply describing your system in natural language. Instead of manually writing UML code, you can interact with the assistant conversationally, and it will generate:

- UML diagram visuals
- The underlying PlantUML code
- A textual description of the diagram

Input formats

You interact with the assistant using **natural language**.

To create a diagram, provide two things in your input:

1. **The story or description** of the system/process you want modeled.
2. **The diagram type** you want generated (choose from: *sequence, use case, class*).

Example:

"I want a use case diagram that shows a customer logging into a shopping website and placing an order."

Input Limitations

- Inputs must be in **written text only**.
- The system does **not** accept images, sketches, or other file formats as input.
- Complex stories may require step-by-step clarifications with the assistant to refine the final diagram.

Outputs

- For every request, the assistant provides three outputs:
 1. **The Diagram** — a generated UML diagram visualization.
 2. **PlantUML Code** — the source code for the diagram, allowing you to edit or reuse it.
 3. **Description** — a textual explanation of the diagram, clarifying what it represents.



Example Workflow

User Input:

"Generate a sequence diagram showing a student logging into a course system and submitting homework."

Assistant Output:

- A sequence diagram visual (actors, actions, interactions).
- The corresponding PlantUML code.
- A description explaining how the diagram captures the student's interaction with the course system.



Notes

- The assistant works best when your descriptions are **clear and concise**.
- You can refine the diagram by asking follow-up questions, such as:
"Add a teacher actor who reviews the homework."
- *The assistant only supports three diagrams: Sequence, Usecase, and Class diagrams*

Thank you!

UML Diagram Generator — README (verbatim)

```
# UML Diagram Generator with Groq & Streamlit
```

A system that converts natural language descriptions into UML class diagrams using Groq's LLM API and displays them via a Streamlit interface.

Features

- ■ Natural Language to UML conversion
- ■ Real-time PlantUML rendering
- ■ Auto-detection of classes, attributes, methods, and relationships
- ■■ Error correction and validation layer
- ■ Responsive web interface

Technologies

- **Backend**: FastAPI
- **Frontend**: Streamlit
- **AI Provider**: Groq (Llama-3-70B)
- **UML Rendering**: PlantUML
- **Environment**: Python 3.10+

Installation

Prerequisites

- Python 3.10+
- Groq API Key (free tier available)

Setup

1. Clone Repository*

```
```bash
git clone https://github.com/yourusername/uml-generator.git
cd uml-generator
```

#### 2. Create Virtual Environment

```
``` bash
python -m venv .venv
````
```

#### 3. Activate Environment

```
``` bash
# Windows
.\venv\Scripts\activate
````
```

#### 4. Install Dependencies

```
``` bash
pip install -r requirements.txt
````
```

#### 5. Configure Environment

```
``` bash
echo "GROQ_API_KEY=your_api_key_here" > .env
````
```

## ## Usage

### ### Running

#### 1. Start Backend API

```
``` bash
cd backend
uvicorn main:app --reload
````
```

...

## 2. Start Frontend UI

```
``` bash
cd frontend
streamlit run app.py
````
```

### ### Example Inputs

Try these system descriptions:

- "Library system with books, members, and loans"
- "E-commerce platform with products, customers, and orders"
- "Hospital management system with doctors, patients, and appointments"

### ## Project Structure

```

```
uml-generator/
    ├── .env
    ├── .venv/
    └── backend/
        ├── main.py      # FastAPI server
        ├── groq_client.py # LLM interface
        ├── plantuml.py   # UML conversion
        ├── schemas.py    # Data models
        └── frontend/
            ├── app.py      # Streamlit UI
            ├── requirements.txt
            └── README.md
```

```