Project Report on

Automatic UML Diagram Generator from Natural Language Descriptions.

Submitted by:

Satyam Shruti Sogattur
Varun Vadlamani

Under the Guidance of:

Dr. Jicheng Fu, Professor in Department of Computer Science

Introduction

This project aims to simplify the process of generating UML diagrams by leveraging the capabilities of a large language model (LLM) and the PlantUML tool. Traditionally, creating UML diagrams is a manual and time-consuming task requiring a good understanding of software modeling. This application allows users to input a simple English description of a system and receive a complete UML diagram as output — without writing any code or UML syntax themselves.

Objective

To build an intuitive tool that:

- Accepts a plain English system description.
- Sends this input to an LLM model to generate a PlantUML-compatible script.
- Processes and renders the diagram using the PlantUML engine.
- Presents the final UML diagram to the user through a clean UI.

System Architecture

The tool operates in the following sequence:

User Input: A user provides a natural language system description through a UI interface (Streamlit).
LLM Interaction: The app sends this input to a locally hosted API (http://csai01:8000/generate/) that wraps around an LLM.
Script Extraction: The response from the LLM is parsed to extract valid PlantUML code.
Script Fixing: Basic sanitization is applied to ensure compatibility with PlantUML standards (e.g., correcting arrow syntax).
Diagram Rendering: The cleaned PlantUML script is saved and passed to the PlantUML tool to generate a .png image of the diagram.
UI Output: The final image is displayed back to the user.

Technologies Used

Component	Technology
Frontend	Streamlit (Python UI)
Backend API	LLM (via HTTP POST request)
Diagram Rendering	PlantUML
Script Processing	Python (regex, subprocess)
Hosting	Local Server

Code Highlights

- extract_plantuml_block(text): Uses regex to isolate the first valid PlantUML code block.
- fix_script(script): Repairs common syntax errors in generated UML scripts.

- get_plantuml_script(prompt): Communicates with the backend LLM server to obtain the UML code.
- Subprocess call to plantuml CLI: Converts the .txt file to an image.

Results and Discussion

The application successfully generated UML diagrams for a variety of system descriptions such as:

- ✓ Library management systems
- ✓ Online shopping platforms
- ✓ Hospital management systems

It abstracts the complexity of writing UML scripts, making the tool useful for:

- → Students learning software design
- → Developers quickly prototyping system designs
- → Analysts visualizing workflows

Conclusion

This project demonstrates a practical and user-friendly approach to converting human language into technical visualizations. It bridges the gap between requirement gathering and design modeling by automating UML diagram generation.