CSCI 578 – Software Architectures – Fall 2025
Individual Assignment #2
Due: *October 9th, 2025*

For this assignment, you will provide (1) a detailed set of functional and non-functional requirements as well as (2) a preliminary architectural model of **PocketLLM Portal**, a lightweight LLM-powered web application designed to run inference on a budget CPU (no GPU required) with a local cache. The system must include both a web frontend and a backend.

Problem Context for PocketLLM Portal

PocketLLM Portal is a lightweight web application that allows end users to interact with a compact language model running entirely on a budget CPU. The system is designed to support resource-constrained environments, so careful use of caching, efficient model serving, and responsive web interactions are essential. PocketLLM Portal meets tight resource constraints typical of entry-level cloud VMs or consumer desktops (*e.g.*, 2–4 vCPUs, ≤8–16 GB RAM). The system should:

- Execute CPU-only inference using a compact model (*e.g.*, 1–8 B parameters, quantized).
- Employ a local cache to reduce repeated computation and outbound calls.
- Offer a browser-based frontend for prompts/results and basic admin tools.
- Provide a backend with model-serving endpoints, caching, and telemetry.

The backend should expose APIs that the frontend and other clients can consume. It should include, but not be limited to, the following:

- Model Serving
- Cache Management
- System Monitoring & Admin Controls

The frontend should provide a web-based interface that allows non-technical users, as well as developers and administrators, to interact with the system:

- Chat Interface
- History & Session Management
- Admin Console
- Developer API Access

Note that the above description is deliberately incomplete. It is intended only as an illustration, which you may choose to or not to use as a starting point for your project. A range of additional reasonable requirements are possible that would add meaningful capabilities to *PocketLLM Portal*.

Also note that you are not expected to train or fine-tune language models as part of this assignment. The purpose of this exercise is to design and reason about the software system architecture that utilizes an existing LLM, not to perform ML research. You should select a publicly available, free lightweight LLM from online sources such as Hugging Face, the

llama.cpp community, etc. Students may also use pre-built Docker images or model runners that wrap such models for CPU inference. The architectural focus should be on how the system orchestrates frontend/backend interactions, caching, and resource constraints, rather than on the LLM technology itself, which is not the scope of this course.

Assignment

You are expected to provide a complete set of functional and non-functional requirements for your PocketLLM Portal website. Note that your website must have both a frontend (running on a web browser) and a backend (running on servers, data stores, or possibly peer devices). You may rely on any online source to derive the requirements, but you must document those sources.

A non-functional requirement is a requirement that relates to the system's non-functional property. A software system's non-functional property (NFP) is a constraint on the manner in which the system implements and delivers its functionality. These include, but are not limited to, efficiency, complexity, scalability, heterogeneity, adaptability, dependability, etc.

You are also required to come up with an architectural model for your PocketLLM Portal website. The model must be in UML, and it must be developed using a UML tool.[1] The model must satisfy all of your requirements. Use as many of the UML diagrams as needed to describe your architecture. You are allowed to select any architectural style(s) and/or pattern(s) that are suitable for your app. In this assignment, you are required to not account for the architectural decisions that the eventual choice of frontend framework (*e.g.*, React) will induce in your solution—in other words, make the frontend architecture independent of the framework choice.

This assignment will form the foundation on which your Assignment #3 (you will be given further requirements and instructions for you to update your architecture based upon) and the collaborative design and implementation exercise (a part of the team project) will be based.

Grading Criteria

You will be graded on the following general criteria:

1. Completeness of your requirements
2. Creativity of your website
3. Completeness of your architectural model
4. Correct use of UML

[1] For example, https://creately.com/lp/uml-diagram-tool.