Project CodEZ
Presentation

SOMA Enterprises

(Stetson B, Owen N, Malcolm Z, Alexander S)

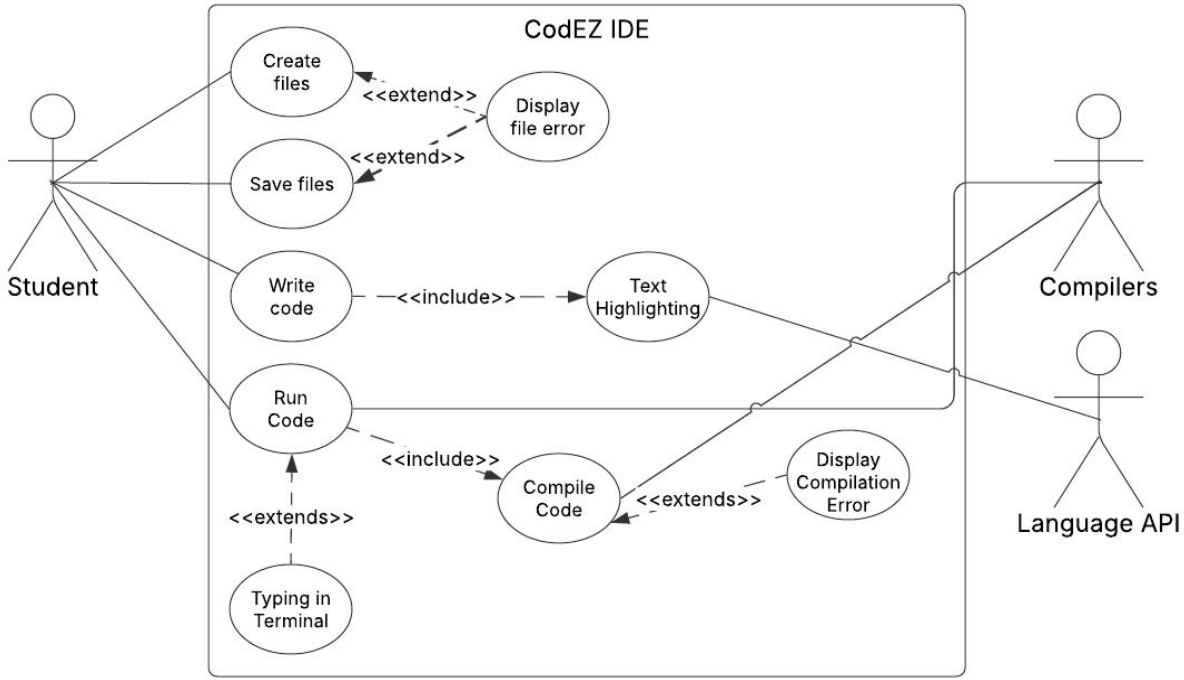
May 2025

CodEZ Business Background

- Goals and Objectives
  - Minimalist Interface
  - Seamless Setup
  - Clear Error Reporting
- Purpose
  - Security Concerns
  - Solution
- Scope
  - Core Functions First
  - Quick Install
  - Performance
- User Characteristics
  - Primary Users (Introductory college/high-school students)
  - Secondary Users (Professors, TAs, tutors, family)

CodEZ IDE Key Features

- Primary Actor:
  - Student
- Interacted Systems:
  - Compilers
  - Language API
- Primary Use Cases:
  - Create and Save files
  - Write code
  - Run code
- Secondary Use Cases:
  - Error Displays
  - Text Highlighting
  - Code compilation
  - Typing in the terminal



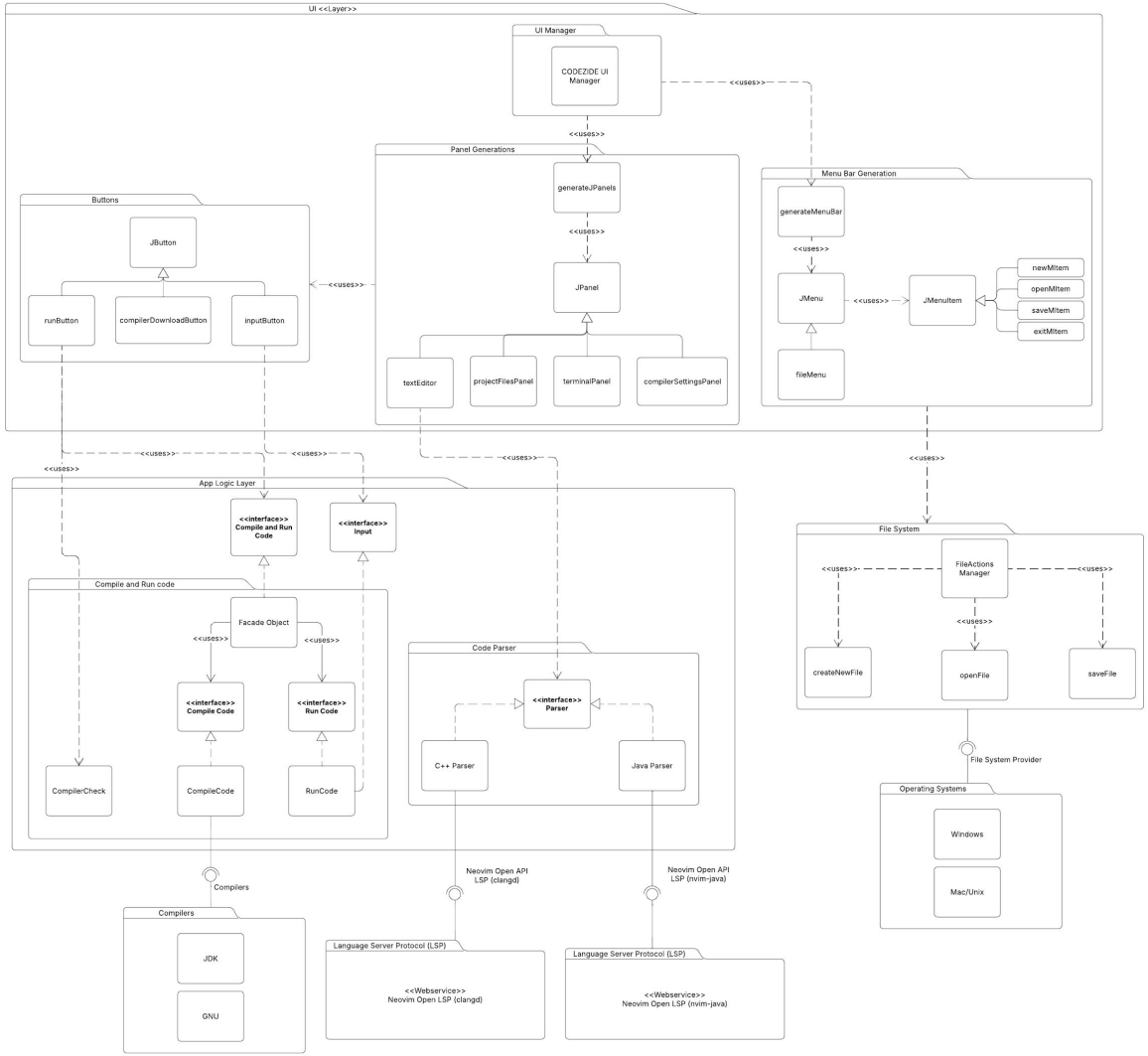
CodEZ IDE Design Concepts

Architecture Design Choice:

- Layered
  - Separation of Concerns

Layers:

- UI layer
- App Logic Layer
  - CompileAndRun package
  - CodeParser package
- File System package
- External Sources
  - Compilers (JDK, GNU)
  - Neovim Open API LSP (clangd, nvim-java)
  - Operating Systems (Windows, MacOS)



CodEZ IDE Design Concepts

Design Patterns:

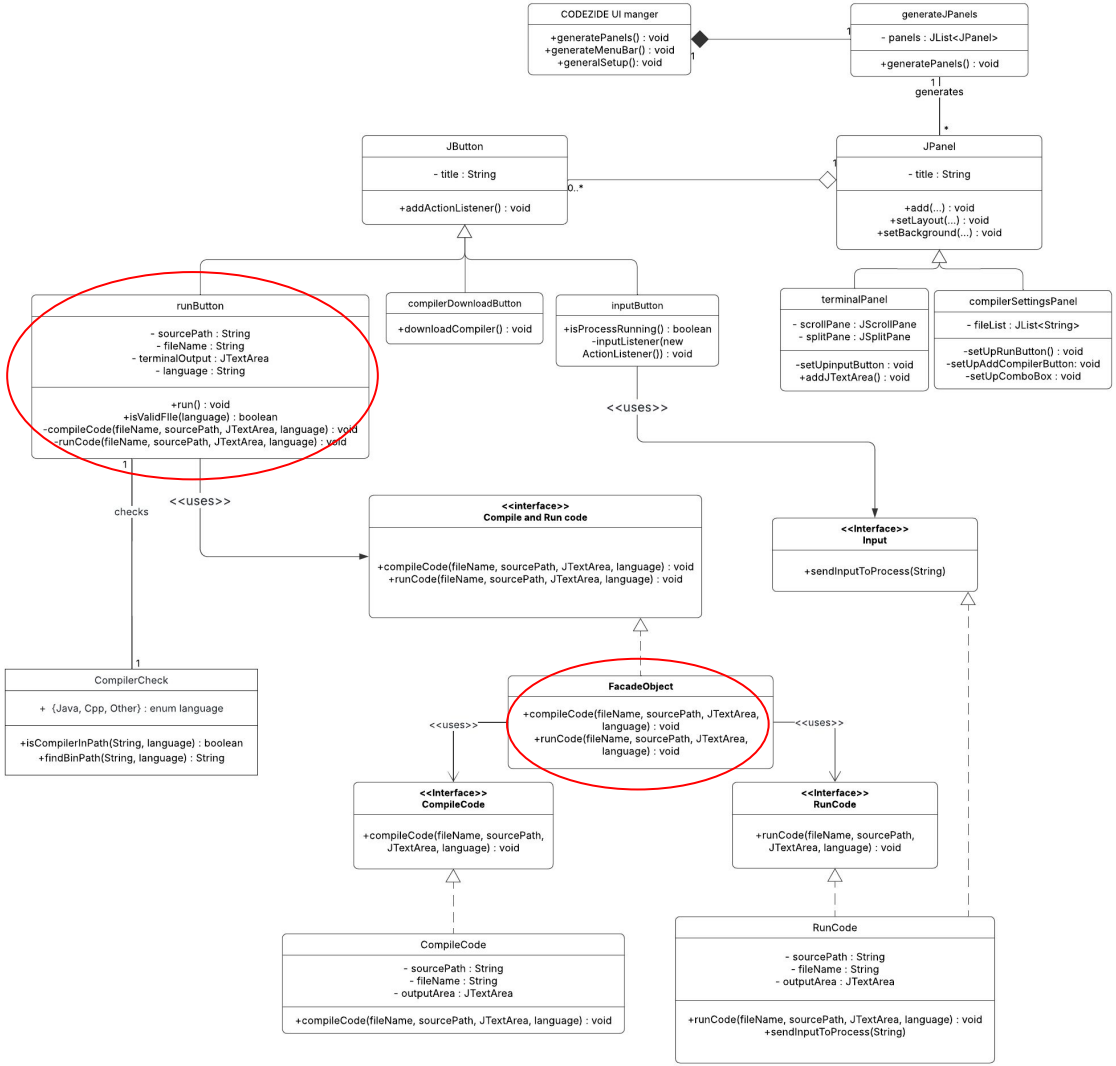
- Inheritance
- Interfaces
- Facade Pattern
- Singleton

Possible Pattern:

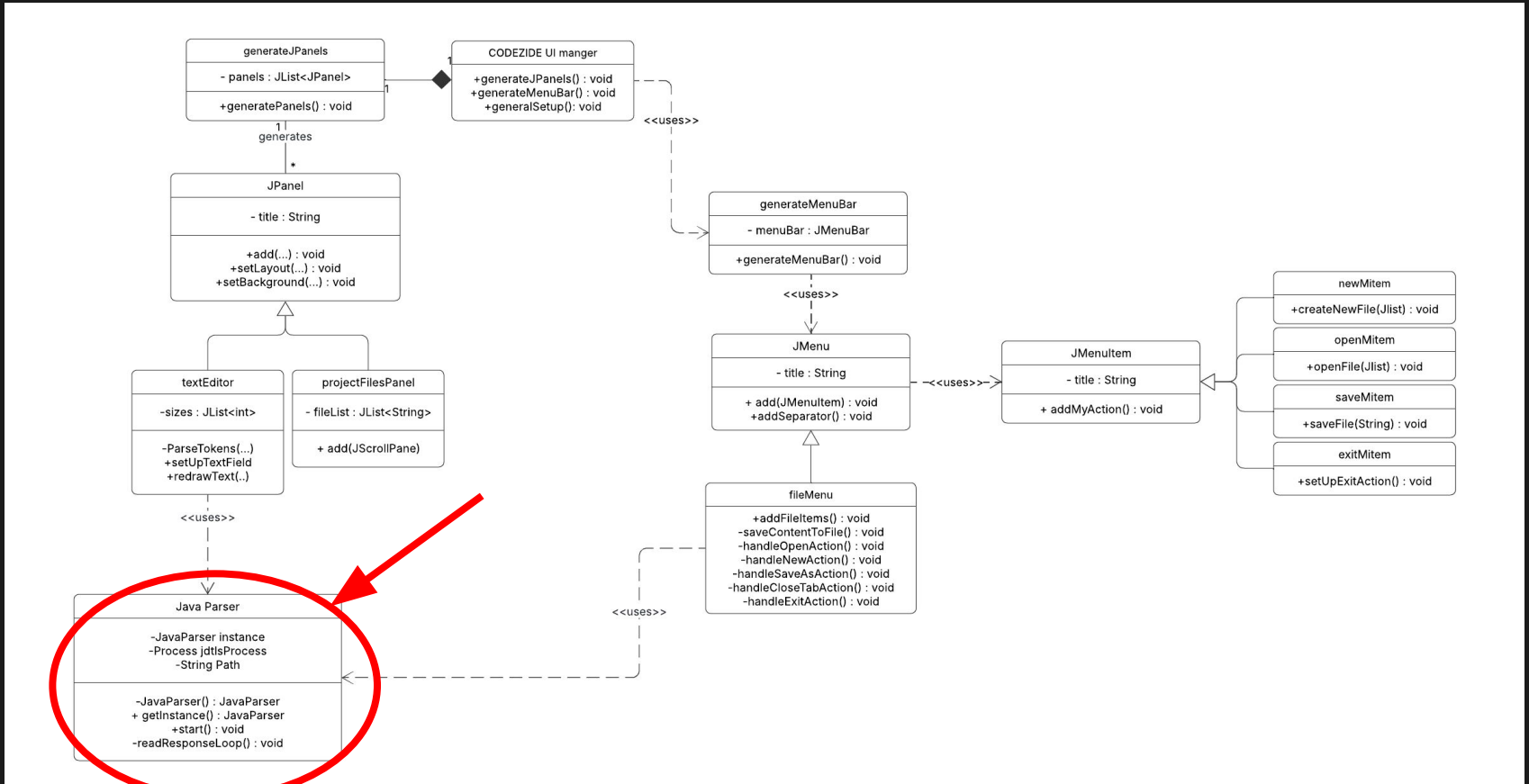
- Strategy

Didn't Use:

- MVC nor DAO



CodEZ IDE Design Pattern: Singleton



CodEZ IDE Behaviour

Key Behaviour

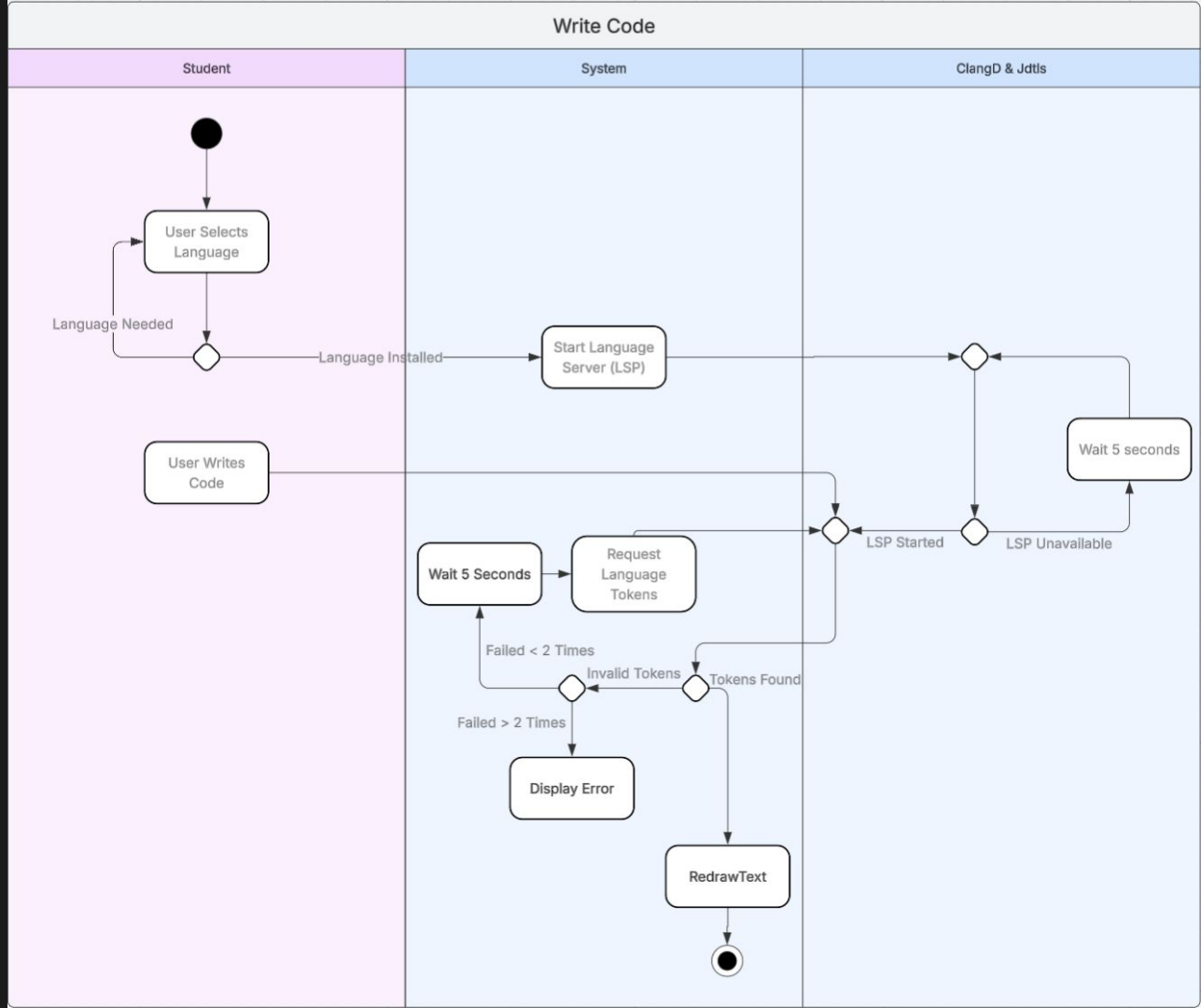
User is able to select their Desired language which begins the LSP

Key Behaviour

When the user writes code the parser will receive the codes and return tokens

Key Behaviour

The System will then Redraw the text with the desired coloring



CodEZ IDE

Demonstrating

https://youtu.be/kgtj6e4F5oo

1. **File System:** Open & Save a file
2. **Text Editing/Viewing:** Type in an opened file see the text formatting
3. **Run File:** Show the file that you saved running
4. **Process Environment (Terminal):** Show input being parsed when submit is clicked

Lessons Learned

Thank you for listening!