



ALGO-MATE

A Design Pattern Insertion Extension for Visual
Studio Code

PROBLEM

- A “Software Design Pattern” is a general, reusable solution to a commonly occurring problem
- Hard to find helpful information on
- Many programmers never learned them
- Easy to find misinformation and distractions online.

WHAT IS ALGO-MATE? - METHODOLOGY

- Algo-Mate is an extension for VS Code which allows users to quickly search for, and insert commonly used design patterns into their code without leaving their IDE.
- The chosen design pattern is added to the file in the proper language based on file extension
- Algo-Mate then assists the user to quickly fill in variable names or insert values into the algorithm through helpful comments and variable grouping.
- If the user needs more information on a design pattern, they can follow our built in help menu to a reputable website.

WHY ALGO-MATE?

- Design pattern snippets across multiple programming languages is a novel idea.
- Algo-Mate is helpful for users who are attempting to implement common design patterns into their code.
- Reduces the need for finding forum-based coding solutions or design pattern documentation, which can quickly draw the user's attention off of their current project.

FEATURES

- Four design patterns implemented in four different languages
- Auto-completion of design patterns for VS Code using JSON snippets
- Navigation to design pattern documentation via UI component
- Generate build data via UI component

TOOLS & TECHNOLOGY

- VS Code and VS Code Extension API
- JSON files include:
 - `cpp.json`
 - `java.json`
 - `javascript.json`
 - `python.json`
- Travis CI
- Git/GitHub
- NPM
- NumPy, Pandas, and Matplotlib modules for Python

The Design Patterns

- Singleton
- Builder
- Adapter
- Observer

ALGO-MATE UI MOCKUP

The screenshot shows the Visual Studio Code editor with a file named 'test.js' open. The code is in JavaScript and demonstrates the Builder design pattern. The word 'builder' is being typed, and a dropdown menu is visible, listing various methods and classes related to the Builder pattern, such as 'builder', 'builderStep1', 'builderStep2', and 'builderGet'. The code includes a 'Shop' class with a 'construct' method and a 'CarBuilder' class with 'step1', 'step2', and 'get' methods.

The first screenshot shows a search bar with the text 'builder' and a dropdown menu listing design patterns: Singleton, Builder, Observer, and Adapter. The second screenshot shows the same search bar with the text 'builder' and a dropdown menu listing programming languages: Cpp, Java, JavaScript, and Python.

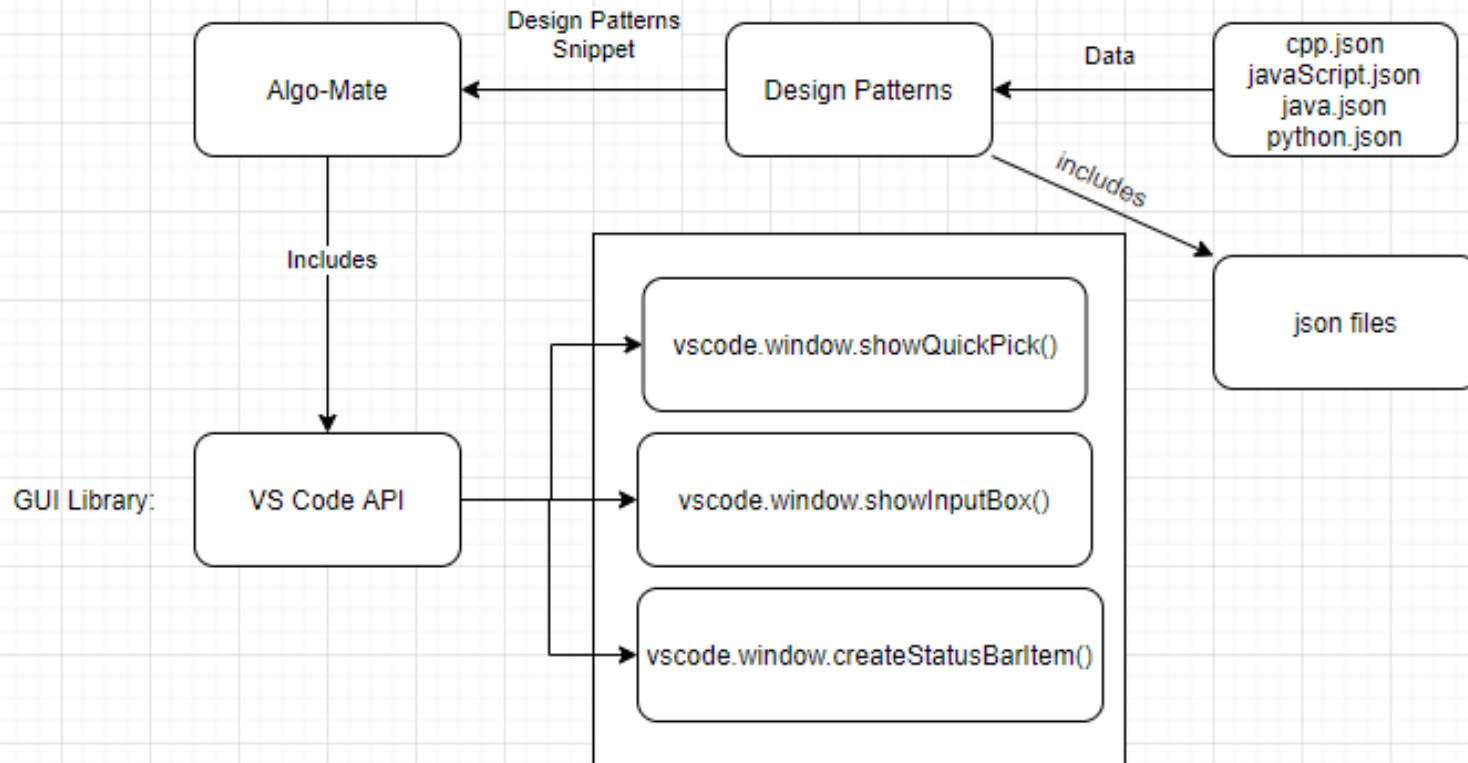
The dialog box is titled 'Visual Studio Code' and contains the following text: 'Do you want Code to open the external website?' followed by the URL 'https://refactoring.guru/design-patterns/builder/java/example'. There are four buttons: 'Open', 'Copy', 'Cancel', and 'Configure Trusted Domains'.

The status bar shows the current branch as 'master', the number of changes as '1', and the current file as '[Algo-Mate]'. There are also icons for navigation and a button labeled '[Algo-Mate Build Data]'.

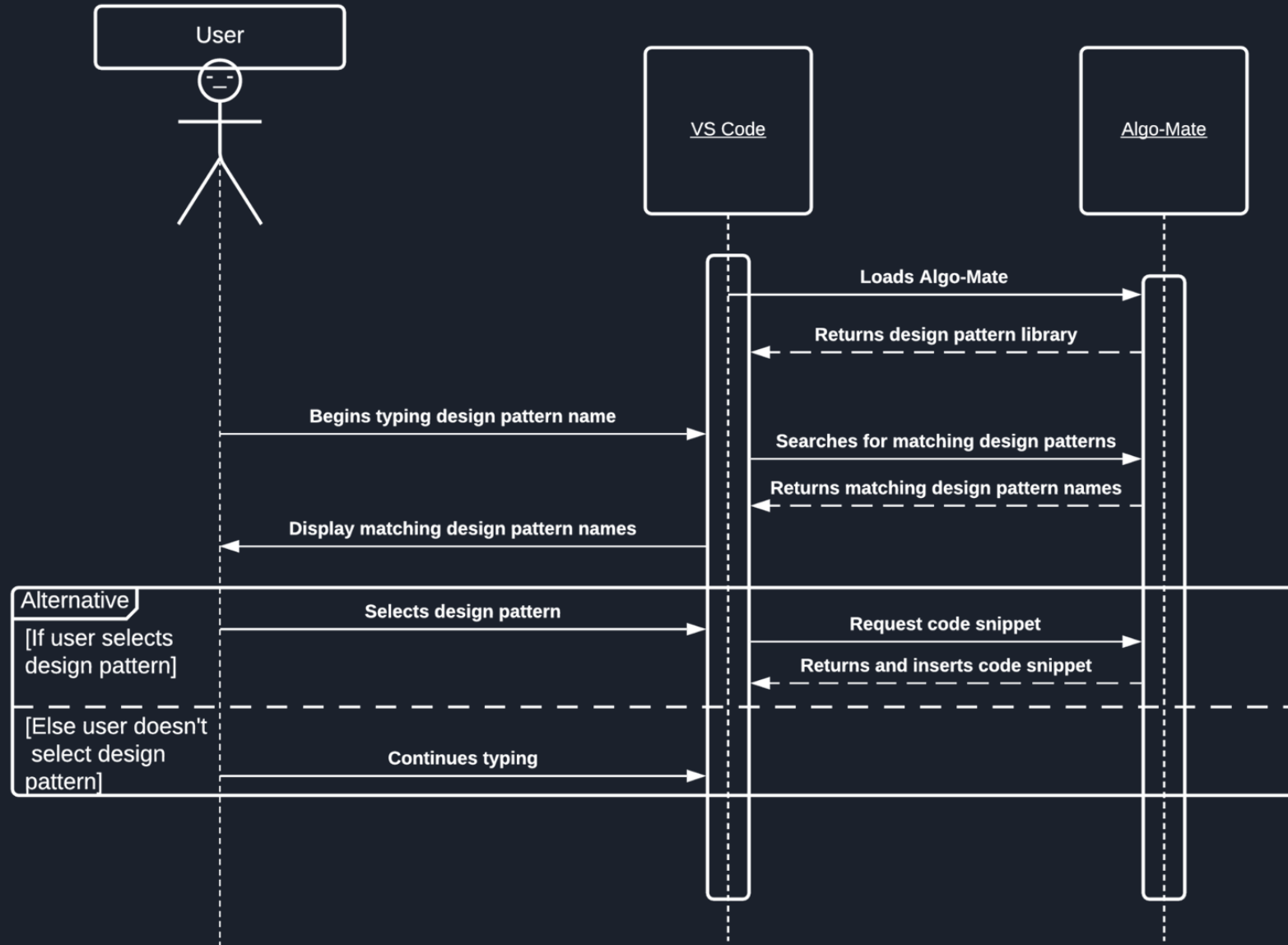
Design Pattern Auto-Completion

Graphical User Interface

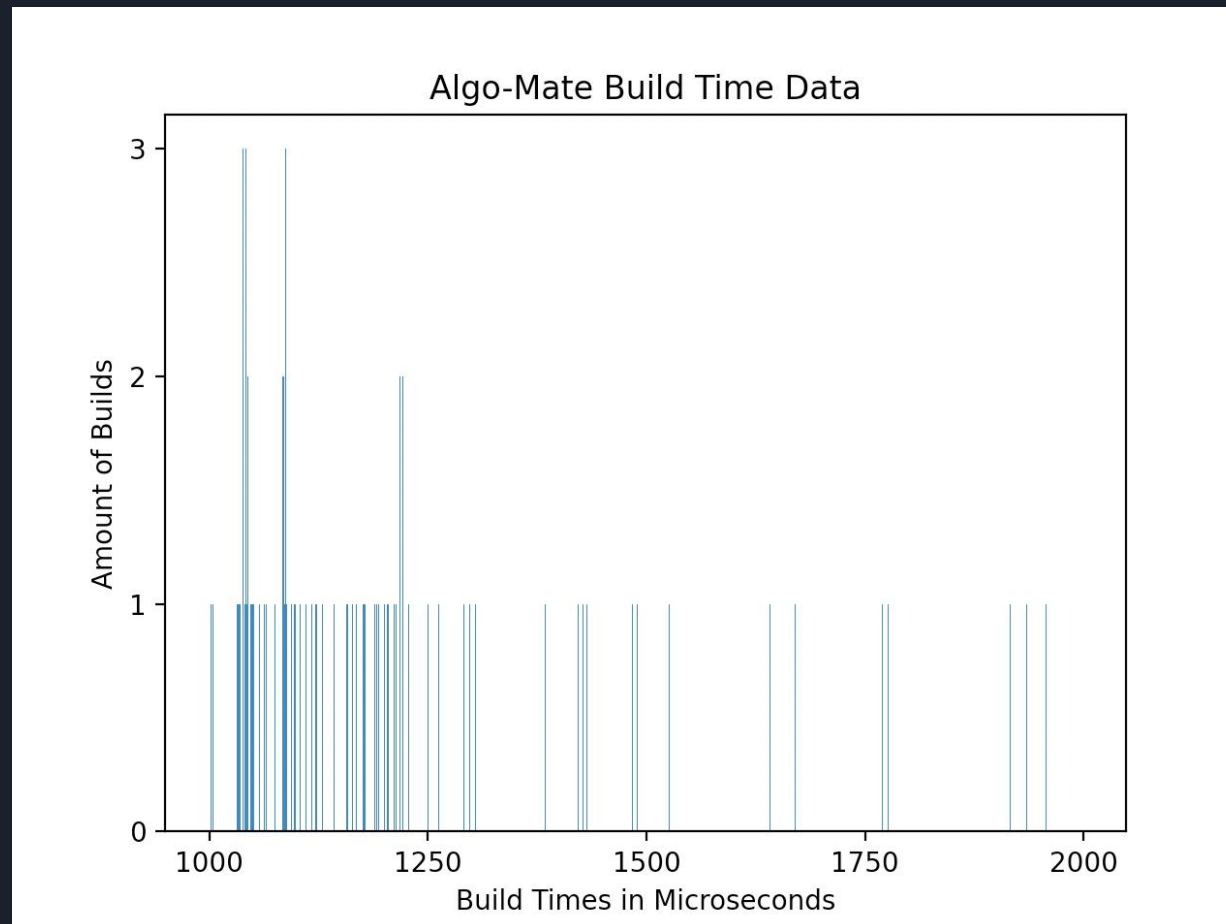
Architecture



User Interaction



RESULTS AND OBSTACLES



DEMONSTRATION

FUTURE WORK

- Add more languages such as C#, Ruby, Go, Swift
- Include more design patterns such as Factory, Prototype, Bridge, Facade, Mediator, and State
- Add more options within the UI
- Include separate design pattern completions with more complete demos/examples

QUESTIONS?