

# Intelligent Code Editor

sdmay21-29 (Spring 2021)

Advisor: Dr. Ali Jannesari

Client: Hung Phan

Team Members: Cory Smith, Jacob Puetz, Benjamin Gonner,  
Evan Christensen, Jordan Silvers

## Introduction

- **Problem:** Programmers new to the field may have a difficult time converting their program ideas to functional code
- **Solution:** An IDE plugin that makes use of a graph neural network text classifier to take their English text and converts it into functional code

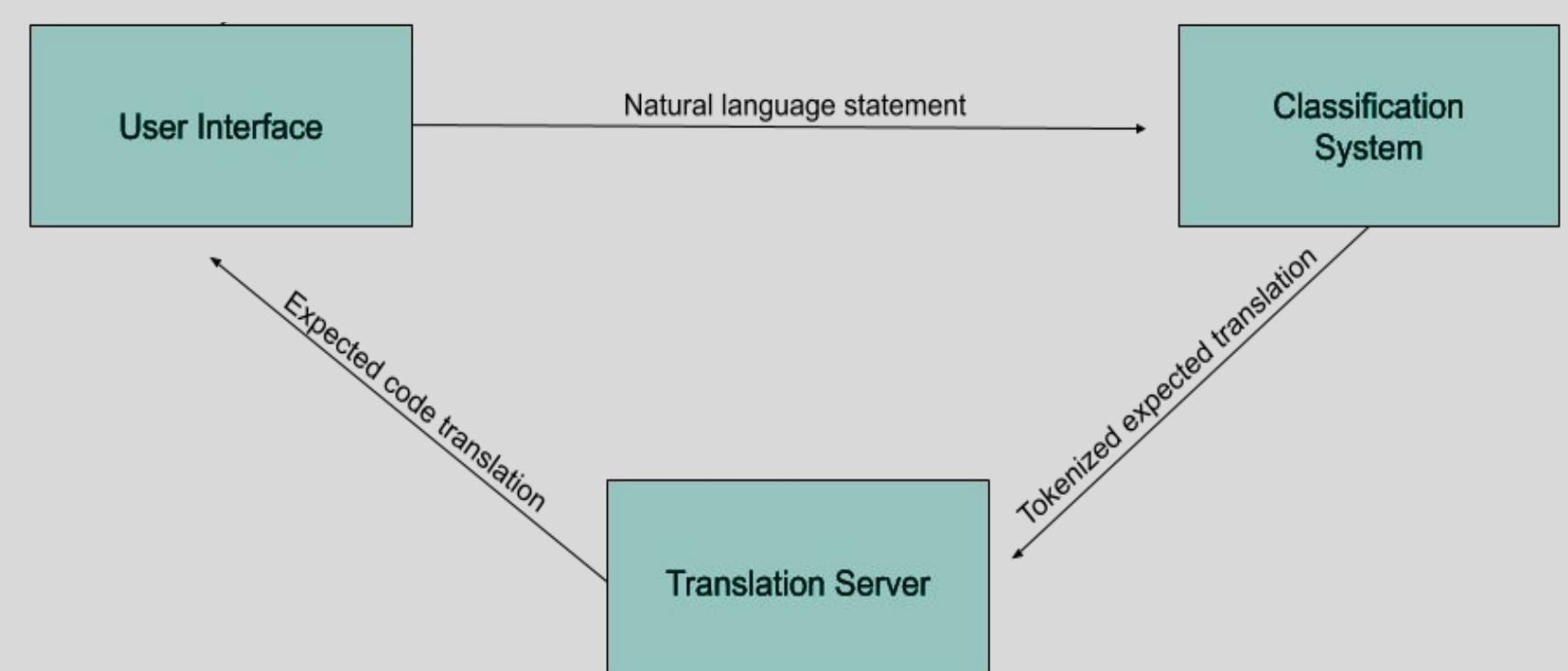
## Intended Users and Uses

- The target audience of this Plugin are people that are inexperienced in programming and want a way to get their ideas to code quickly
- Specifically, the IDE is intended to take Natural English and convert it into C++.



## Design Requirements

- **Functional Requirements:**
  - Ability for user to input code and natural language into an editor
  - Ability for plugin to classify and translate natural language to code
  - Translation should only be initiated by the user
  - Code must be executable once translated
- **Non-functional Requirements:**
  - Accuracy over 50%
  - Translation process should be quick
  - The plugin should be easy to use once installed
- **Engineering Constraints:**
  - Must be completed in two semesters
  - Completed with only currently available resources
- **Operating Environment:** A plugin to Visual Studios and a Python server hosted on a windows machine
- **Standards Used:** IEEE P14764, IEEE 29119-2-2013, IEEE P15026-2, Agile Workflow, TDD

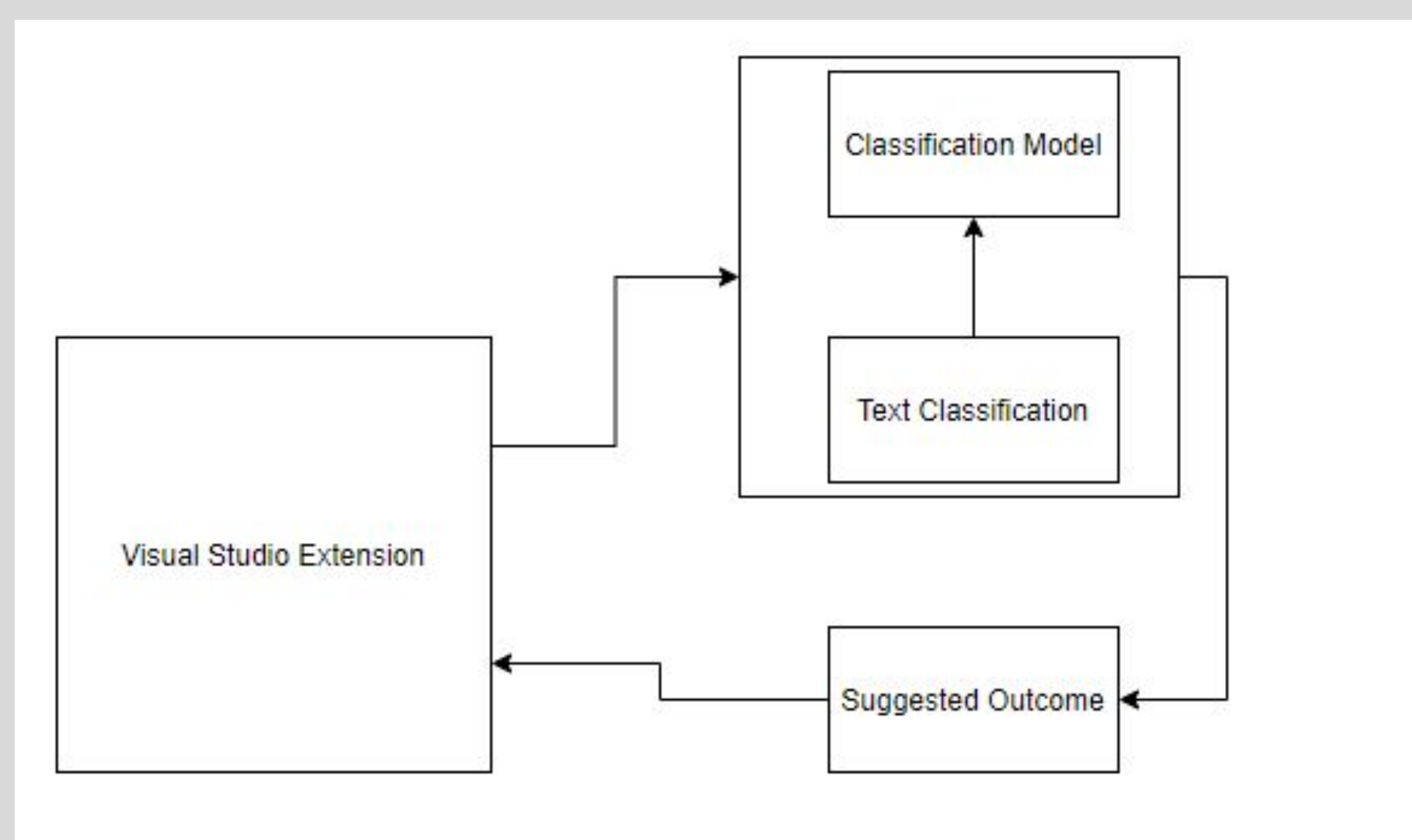


## Design Approach

- **User Interface - Visual Studio Extension**
  - Highlight line of desired code
  - Right-Click to access extension to translate line
  - Changes desired line to expected code
- **Training Data**
  - SPoC Dataset from Stanford University
  - Contains roughly 300,000 lines of pseudocode and equivalent code
- **Preprocessing**
  - Uses regex to parse code
  - Determines classification based on equivalent pseudocode from dataset

## Technical Details

- Server created in Python for classification
- Pre-processing of training data is done in Python
- The C# plugin for Visual Studios moves user input to and from the server and replaces text
- Programming languages: C# and Python
- Libraries: TensorFlow, Regex
- Environment/Tools: Visual Studios



## Testing

- Used regex101 for pre-processing testing
- Accuracy based on TensorFlow testing of 10% of validation data
- **Accuracy:** 50-60%