Joshua Rechkemmer, Adrian Cordona
Lee CSCI 4316-001
9/15/22

Source Code to UML Diagram Tool
Needs Statement

Purpose:

This document provides the functional need statements for program analyzer. This project is designed to give student teams experience in developing intermediate-sized software product. Although subsets of this exercise can be implemented with a relatively modest amount of code, even experienced students will find this exercise to be challenging and realistic examples of the type of software development work they will encounter as working software engineers. The principal objective of this exercise is to provide challenging programming problems and to provide experience with sound engineering and team practices.

Source Code to UML Tool Functional Need Statement

The source code to UML tool is an application tool that a programmer can use to create UML diagrams by analyzing a source code project and then drawing it.

1. **Source Code to UML Functions**
    i. Analyze an existing program written in Java to determine the characteristics of each program file
    ii. Discover the connections between each class in the project
    iii. Produce a formatted file containing the attributes of all of the classes in the given project
    iv. Draw a UML diagram from the created formatted file
    v. Have an interface, console or graphical, for the user to use
    vi. The created UML diagram should be saved onto an image file type
2. **Tool Specifications**
    i. Class Specifications that will be analyzed
        i. Class names and their types (i.e. interface)
        ii. Function headers with their return types, function attributes, name, and parameters
        iii. Class attributes is the variable instance data
        iv. Relationship to other class in the project
    ii. UML Specifications
        i. Class Name
        ii. Class Type
        iii. Class Attributes
        iv. Class Functions
        v. Above Information is contained in a shape
        vi. Correct arrow pointing to other class depending on relationship type
    iii. Ignore comments and blank lines

3. Documentation Specification

i. Tool Documentations will describe the architecture and design of the program in detail

ii. User Manual containing information on installation and operation must be provided

iii. Test Results must satisfy all of the requirements listed in *Section 2 Tool Specifications*