**Abbottabad University of Science & Technology**

**SOFTWARE REQUIREMENTS SPECIFICATION   
(SRS DOCUMENT)**

**For**

**<Text Editor App Using Stacks>**  
 Version 1.0

***By***

**Haroon Imran FA-14723**

**Supervisor**

**(Sir Jamal Abdul Ahad)**

Table of Contents

[**1.** **Introduction** 3](#_Toc185955585)

[**1.1 Purpose** 3](#_Toc185955586)

[This document specifies the requirements for a Python-based Code Editor application. It outlines the system’s functional and non-functional requirements, operating environment, and user interactions. The application aims to provide developers with an efficient and feature-rich platform for coding, debugging, and managing stacks within a clean and user-friendly interface. 3](#_Toc185955587)

[**1.2 Document Conventions** 3](#_Toc185955588)

[**1.3.1 Scope Definition** 3](#_Toc185955589)

[The Code Editor application provides a platform for writing, editing, and managing Python code with integrated stack operations and undo/redo functionality. 3](#_Toc185955590)

[**1.3.2 Core Features** 3](#_Toc185955591)

[ Syntax-highlighted code editor for Python. 3](#_Toc185955592)

[ Undo and redo operations with stack-based tracking. 3](#_Toc185955593)

[ Tabbed interface for editing multiple files. 3](#_Toc185955594)

[ Save functionality for stacks and editor content. 3](#_Toc185955595)

[ User interface for managing stacks with operations like append, pop, peek, and size. 3](#_Toc185955596)

[**1.3.3 Subsequent Releases** 3](#_Toc185955597)

[**1.3.4 Alignment with User and Business Goals** 4](#_Toc185955598)

[ Python Documentation: https://docs.python.org/ 4](#_Toc185955599)

[ Tkinter Documentation: https://docs.python.org/3/library/tkinter.html 4](#_Toc185955600)

[ Tkcode Library: https://github.com/tkcode 4](#_Toc185955601)

[**2. over all Description** 4](#_Toc185955602)

[**2.1 Product Perspective** 4](#_Toc185955603)

[**2.1.1 Product Context** 4](#_Toc185955604)

[The application is a standalone desktop software designed for individual developers. 4](#_Toc185955605)

[**2.1.2 Product Origin** 4](#_Toc185955606)

[Inspired by the need for lightweight, Python-focused code editors with stack management capabilities. 4](#_Toc185955607)

[**2.2 User Classes and Characteristics** 4](#_Toc185955608)

[**2.3 Operating Environment** 5](#_Toc185955609)

[**3. System Features** 5](#_Toc185955610)

[**4. External Interface Requirements** 6](#_Toc185955611)

[**4.1 User Interfaces** 6](#_Toc185955612)

[**4.2 Software Interfaces** 6](#_Toc185955613)

[**4.3 Hardware Interfaces** 6](#_Toc185955614)

[**5. Quality Attributes** 6](#_Toc185955615)

[**5.1 Performance** 6](#_Toc185955616)

[**5.2 Reliability** 6](#_Toc185955617)

# **Introduction**

## **1.1 Purpose**

## This document specifies the requirements for a Python-based Code Editor application. It outlines the system’s functional and non-functional requirements, operating environment, and user interactions. The application aims to provide developers with an efficient and feature-rich platform for coding, debugging, and managing stacks within a clean and user-friendly interface.

### **1.2 Document Conventions**

* Python conventions such as PEP 8 are followed for the code.
* UI elements are specified using Tkinter nomenclature.
* External libraries are referenced by their Python package names (e.g., tkcode).

#### **1.3 Project Scope**

### **1.3.1 Scope Definition**

### The Code Editor application provides a platform for writing, editing, and managing Python code with integrated stack operations and undo/redo functionality.

### **1.3.2 Core Features**

### Syntax-highlighted code editor for Python.

### Undo and redo operations with stack-based tracking.

### Tabbed interface for editing multiple files.

### Save functionality for stacks and editor content.

### User interface for managing stacks with operations like append, pop, peek, and size.

### **1.3.3 Subsequent Releases**

Future updates will include:

* Support for additional programming languages.
* Advanced debugging tools.
* Cloud-based synchronization and collaborative editing.

.

### **1.3.4 Alignment with User and Business Goals**

The application supports developers’ productivity and learning by providing essential tools for code management and stack operations within an accessible desktop interface.

##### **1.4 References**

# Python Documentation: <https://docs.python.org/>

# Tkinter Documentation: <https://docs.python.org/3/library/tkinter.html>

# Tkcode Library: <https://github.com/tkcode>

# **2. over all Description**

## **2.1 Product Perspective**

### **2.1.1 Product Context**

### The application is a standalone desktop software designed for individual developers.

### **2.1.2 Product Origin**

### Inspired by the need for lightweight, Python-focused code editors with stack management capabilities.

**2.1.3 Product Relationship to Existing Systems**

Integrates with local file systems for saving and loading files but operates independently of cloud-based ecosystems.

**2.1.4 Product Ecosystem**

Provides an extensible framework for adding new features such as language support and plugins.

## **2.2 User Classes and Characteristics**

**2.2.1 Tech Enthusiasts**

Users familiar with programming and interested in lightweight tools.

**2.2.2 Casual Shoppers**

Non-programmers exploring coding.

**2.2.3 Favored User Class**

Tech enthusiasts and beginner developers are the primary user groups.

**2.2.4 Alignment with User Needs**

Focused on ease of use, intuitive interface, and essential coding tools.

### **2.3 Operating Environment**

**2.3.1 Hardware Platform**

Minimum requirement: 2GB RAM, Intel i3 processor, and 50MB free disk space.

**2.3.2 Operating Systems and Versions**

Supports Windows 10 and later, macOS 10.15 and later, Linux distributions with Python 3.8+

#### **2.4 Design and Implementation Constraints**

**2.4.2 Database Technology**

SQLite3 for saving stack-related data.

**2.4.4 Third-Party Integrations**

* tkcode library for enhanced code editing features.
* ttk widgets for modern UI elements.

**2.4.5 User Interface Design**

Follows minimalistic design principles for usability and performance.

##### **2.5 Assumptions and Dependencies**

**2.5.1 Assumptions**

* Users have Python installed on their systems.
* Users have basic familiarity with file operations.

**2.5.2 Dependencies**

* Python 3.8 or higher.
* Required libraries: tkinter, tkcode, ttk.

# **3. System Features**

1. Syntax highlighting.
2. Undo and redo functionality.
3. Stack management UI.
4. File saving/loading for stack content.

# **4. External Interface Requirements**

## **4.1 User Interfaces**

**4.1.1 Design Standards and Guidelines**

Adheres to Tkinter’s layout standards.

**4.1.2 Screen Layout and Resolution**

Responsive design for 1366x768 and higher resolutions.

**4.1.3 Standard Interface Elements**

Utilizes ttk buttons, labels, and frames for consistency.

### **4.2 Software Interfaces**

Integrates with file dialog for saving and loading data.

**4.2.4 Non-Functional Requirements**

Fast load times under 2 seconds.

### **4.3 Hardware Interfaces**

Basic I/O operations through local devices.

**4.3.1 Supported Device Types**

Desktop PCs and laptops.

# **5. Quality Attributes**

## **5.1 Performance**

Efficient operation with minimal resource consumption.

### **5.2 Reliability**

Ensures content recovery via undo/redo.

#### **5.3 Usability**

User-friendly interface with tooltips and error handling.

##### **5.4 Security**

Limited access to system files.

###### **5.5 Maintainability**

Codebase modularity for future enhancements.