Faculty of
Engineering Ain
Shams University Credit
hours program

CSE 426 - Software Maintenance and Evolution

Assignment (1)

SRS

Submitted to: Prof. Ayman Bahaa-Eldin
Submitted by: Mariam Hassan Nassar
ID: 17P6075

Brief Description

Anubis-IDE is an open-source desktop text editor that helps provide a simple integrated development environment to write, edit, run and compile python code on microcontrollers. This project is supposed to facilitate development for embedded-systems engineers. It will help them compile, build and run their code directly on the microcontroller as efficiently as possible.

System Requirements

A. Functional Requirements

The software must be able to:

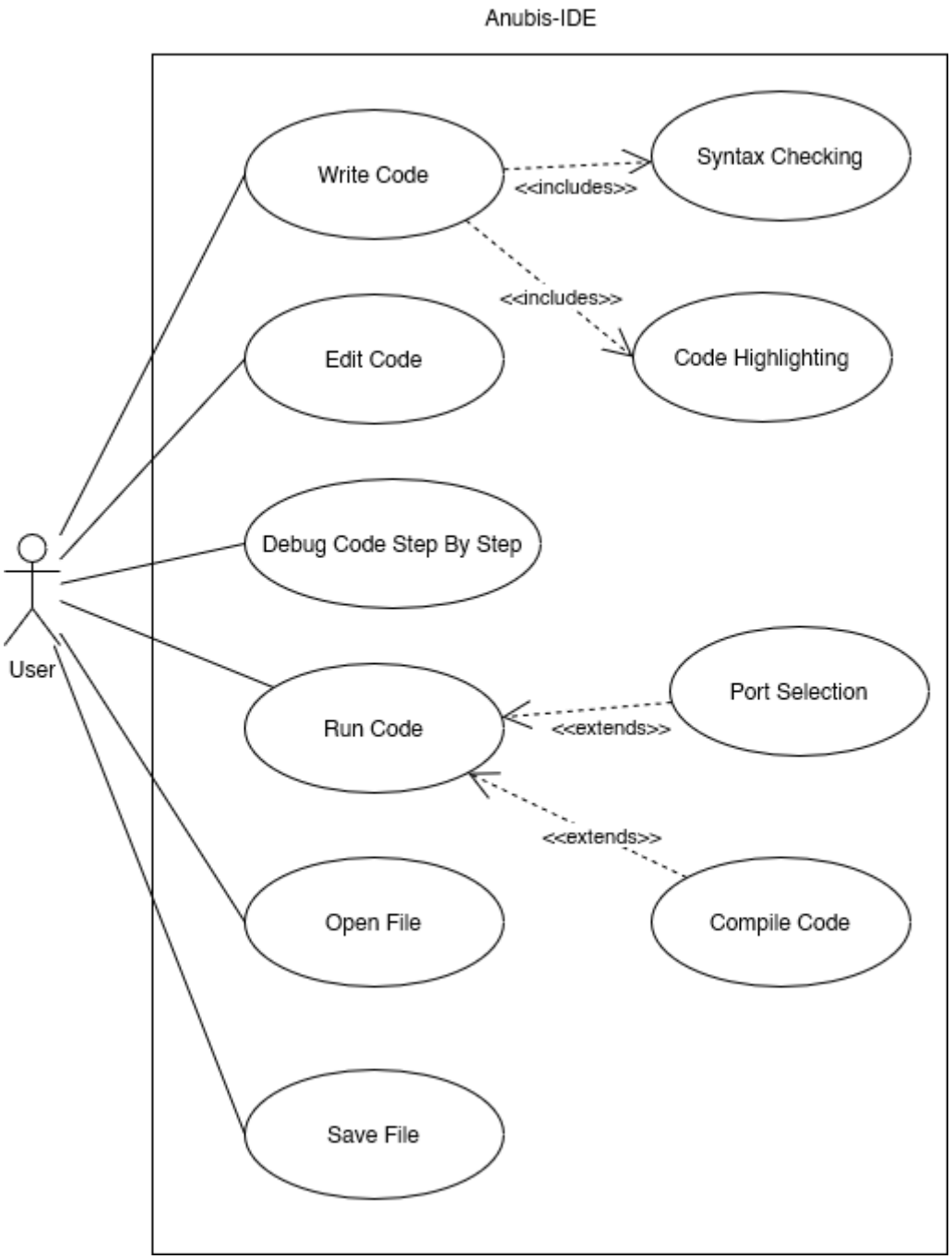
1) Support opening and editing any text files
2) Allow writing micro-python codes to files
3) Support code highlighting
4) Support syntax checking
5) Support auto-completion
6) Provide debugging tool
7) Provide list of all available ports to select one
8) Allow selection of attached microcontroller port before run & compile
9) Compile, flush and run code on selected microcontroller
10) Have panel to display class hierarchy
11) Have panel to display project structure in files and folders directory format
12) Have code editor panel with code highlighting for reserved words, comments & variables
13) Save files in currently opened directory

B. Non-Functional Requirements

The software must:

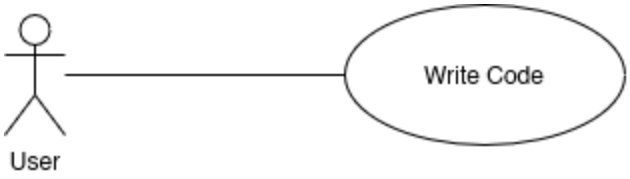
1) Be written in python
2) Be compatible with different operating systems
3) Provide error detection and feedback within a second of request
4) Not exceed 4GB of RAM usage
5) Use Git for version control
6) Have a public repo on GitHub
7) Follow agile process model
8) Be delivered within 3 main releases that are 3 months-long each
9) Declare dependencies
10) Provide installation instructions

Use Case Diagram



Use Case Description

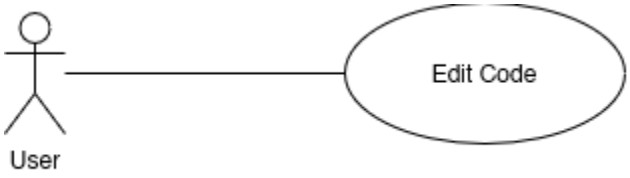
1) Write Code



Description: User should be able to write python code in text-editing panel
Primary Actor: User
Main Flow:

1. User opens IDE
2. User starts typing in text-editing panel

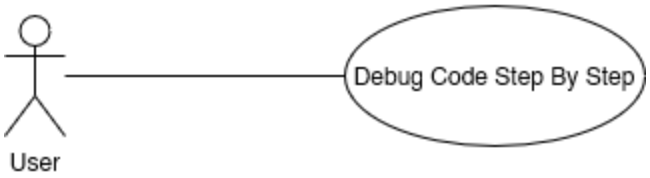
2) Edit Code



Description: User should be able to edit existing python code in text-editing panel
Primary Actor: User
Main Flow:

1. User selects python file from tree view
2. Code opens in text-editing panel
3. User starts editing code

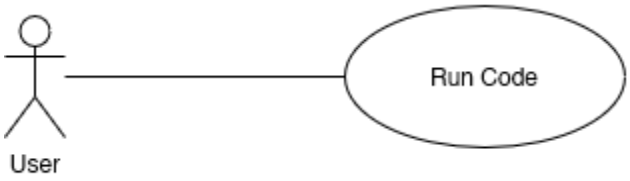
3) Debug Code Step By Step



Description: User should be able to use break points to debug python code
Primary Actor: User
Main Flow:

1. User selects line of code and adds break point to it
2. User specify running port
3. User runs code
4. New panel appears to show used variables, values should appear to user

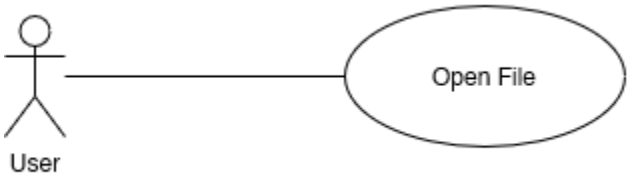
4) Run Code



Description: User should be able to run python code on a microcontroller
Primary Actor: User
Main Flow:

1. User specify running port
2. User chooses to run code
3. Feedback message should appear to user showing if run is successful or not

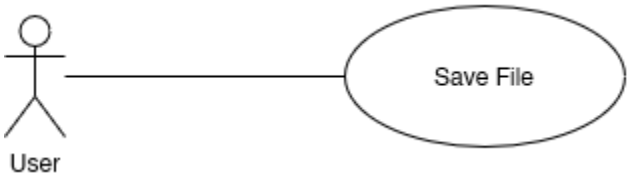
5) Open File



Description: User should be able to open existing code file from the tree view or file menu
Primary Actor: User
Main Flow:

1. User opens file menu
2. User selects “Open”
3. User selects desired file through the file explorer

6) Save File



Description: User should be able to save current progress
Primary Actor: User
Main Flow:

1. User opens file menu
2. User selects “Save”