# Keeping Functions In Line

https://github.com/jilliebean/Senior-Project.git

Jillian David j_david1@u.pacific.edu

# Motivation for the System

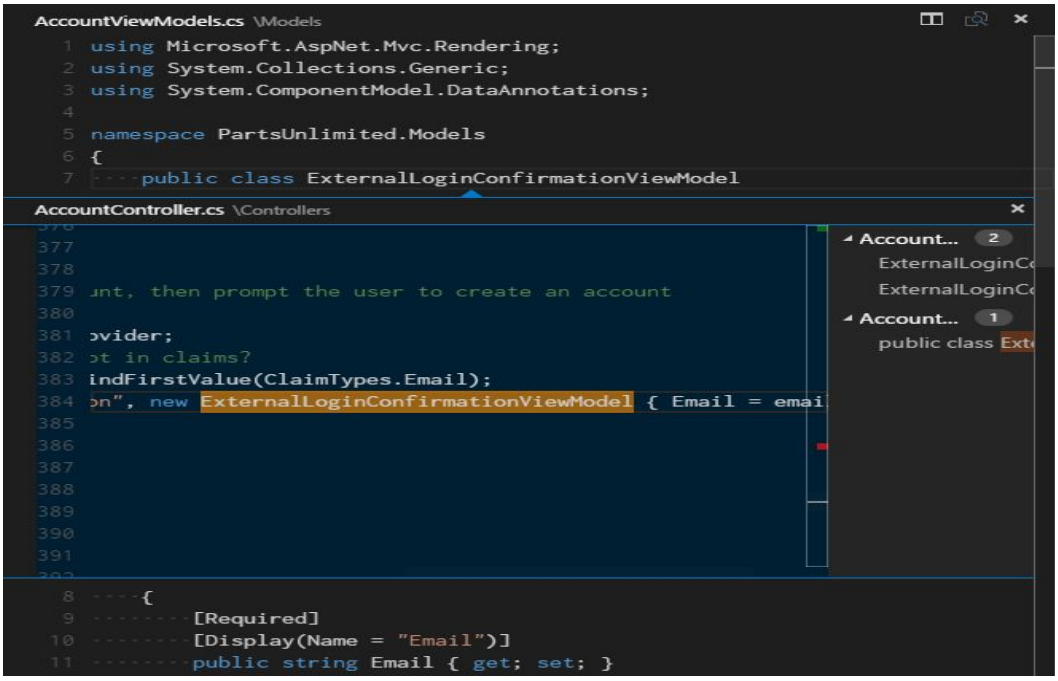
Eclipse is one of the most popular integrated development environments (IDE) programmers utilize. “Keeping Functions In Line” is an Eclipse plug-in that will allow programmers to seek, edit, and save functions without the added hassle of navigating and eventually finding the function they wish to edit.

# The Solution

Companies and programmers alike must use their time efficiently and searching for a function to edit is not conducive to this ideology. In an attempt to make coding more efficient, “Keeping Functions In Line” will implement a peek and edit function plug-in. This Eclipse plug-in will allow users to edit a function not in sight, save the edit, and continue navigation where they currently were.

# The Current Solution

“Keeping Functions In Line” has been seen before. There is an inline function editor within Visual Studio, but Visual Studio does not readily support Java and J#. Visual Studio requires 2.3 Gigabytes of storage and 250 Megabytes of memory, whereas Eclipse only requires 360 Megabytes of storage and 3 Megabytes of memory; thus, for some, Eclipse is a better solution. By making this plug-in for Eclipse, “Keeping Functions In Line” offers the same  inline editing feature so many like about Visual Studio without the overhead.




An example of Visual Studio’s inline function editing

# Project Stakeholders

### *Product Owner:*

The owner of this plug-in will be the Eclipse community. Eclipse is an open source IDE, which means anyone is able to edit Eclipse's functionality, and add functionality. The plug-in will be hosted by Eclipse and will thus also be owned by the IBM Software Group.

### *Client / User:*

The client for "Keeping Functions In Line" is the entire Eclipse community. Eclipse is an open source integrated development environment which allows its users to create, share, and edit existing code. The client will be a programmer looking to work efficiently within the Eclipse environment.

# Typical System Users

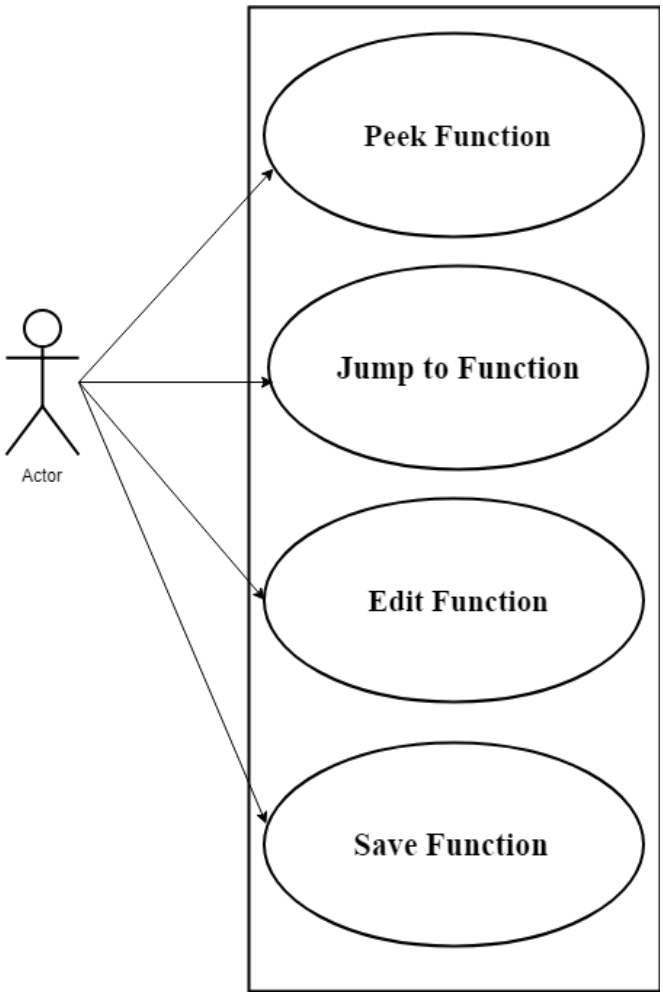
The typical user will most likely be a programmer. This user can be a beginner, intermediate, or well-seasoned programmer. This user will be a typical Eclipse programmer and someone who works in this integrated development environment regularly on large projects that make navigation difficult.

# Non-Functional Requirements

- *Hot Keys*
  The system will be able to receive function keys that allow the user to instantly jump to, edit, and save function they're working on

# Functional Requirements

## Use Case Diagram




## Casual Use Case Descriptions

User Use Cases:

1. Peek Function: Users will click or use hot keys to open up “Keeping Function’s In Line” where the function they are cursored over will appear in a small textbox window on the bottom right hand of the screen. This will simply display the function within the small window.
2. Jump to Function: Users will click or use hot keys to jump straight to the function they have cursored over. THis will not utilize the small textbox window, rather it will navigate for the user to the function they seek.

3. Edit Function: Users will click of use hot keys to open “Keeping Functions In Line” where the function they are cursored over will appear in a small textbox window in the bottom right hand of the screen. The user will then be able to cursor over the function within the window and begin adding or deleting lines code code as they see fit. The exit window tab in the top right will change to a circle indicating something has changed that needs to be saved prior to exiting the program.
4. Save Function: Users will click or use hot keys to save and close “Keeping Functions In Line”. The user will have already had the plug-in open and have been editing functionality in their code. The user will be able to quickly save and the window will close.

## Fully-Dressed Use Case Descriptions

Peek Function

1. User cursors over function they wish to peek
    a. User presses defined hotkey, or opens “Keeping Functions In Line” plug-in
        i. User scrolls through small window to read the function
            1. User closes the window using the exit button in the top right corner
            2. User closes the window by pressing the escape button

Jump to Function

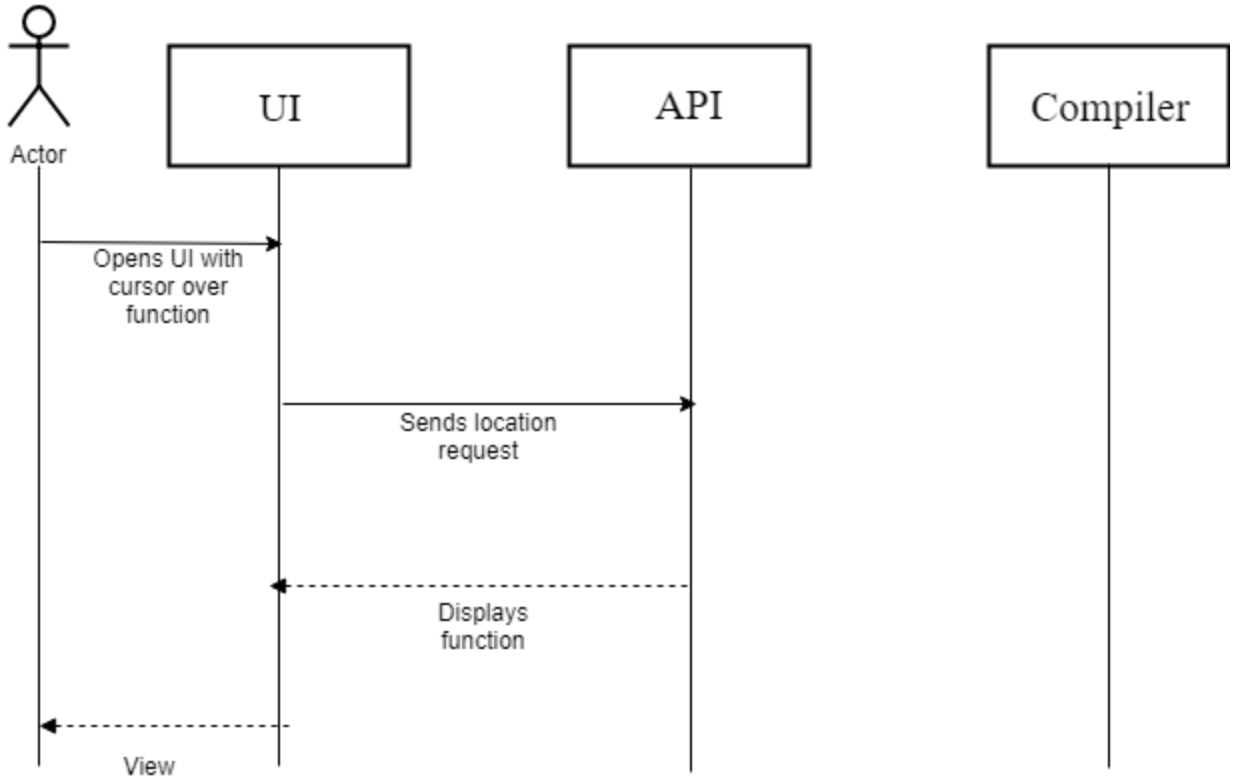
1. User cursors over function they wish to jump to
    a. User presses hotkey, or opens “Keeping Functions In Line” plug-in
        i. After jumping to function, the user can edit the function normally as specified by Eclipse

Edit Function

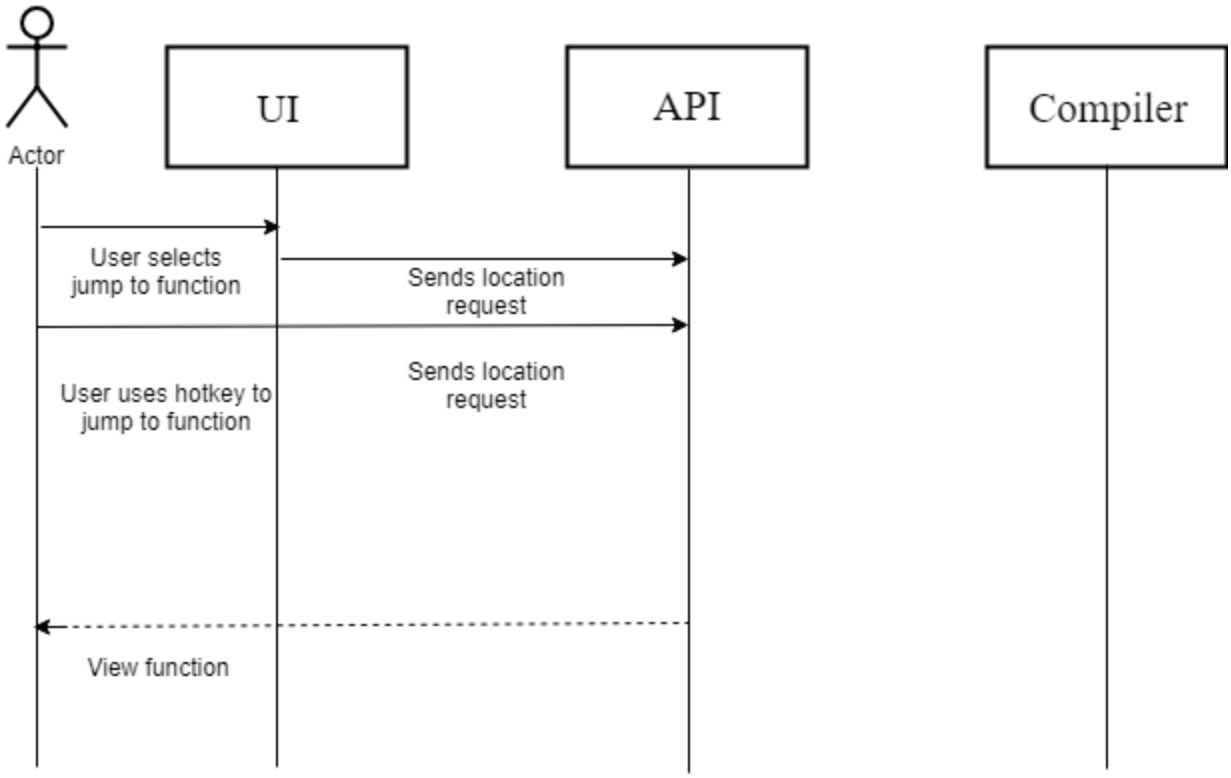
1. User cursors over the function they wish to edit
    a. User presses hotkey or opens “Keeping Functions In Line” plug-in
        i. User cursors over text within window
            1. User begins typing and editing the function they wish to edit
                a. User presses hotkey to save function
                    i. User escapes or exits “Keeping Functions In Line” plug-in
                b. User exits the function but is prompted of their unsaved work
                    i. User decides to save and exit
                    ii. User decides to not save and exit
                    iii. User decides to cancel request and stay in the plug-in environment
        ii. User uses hotkey to jump to function in their workspace
        iii. User simply peeks function

# System Sequence Diagrams

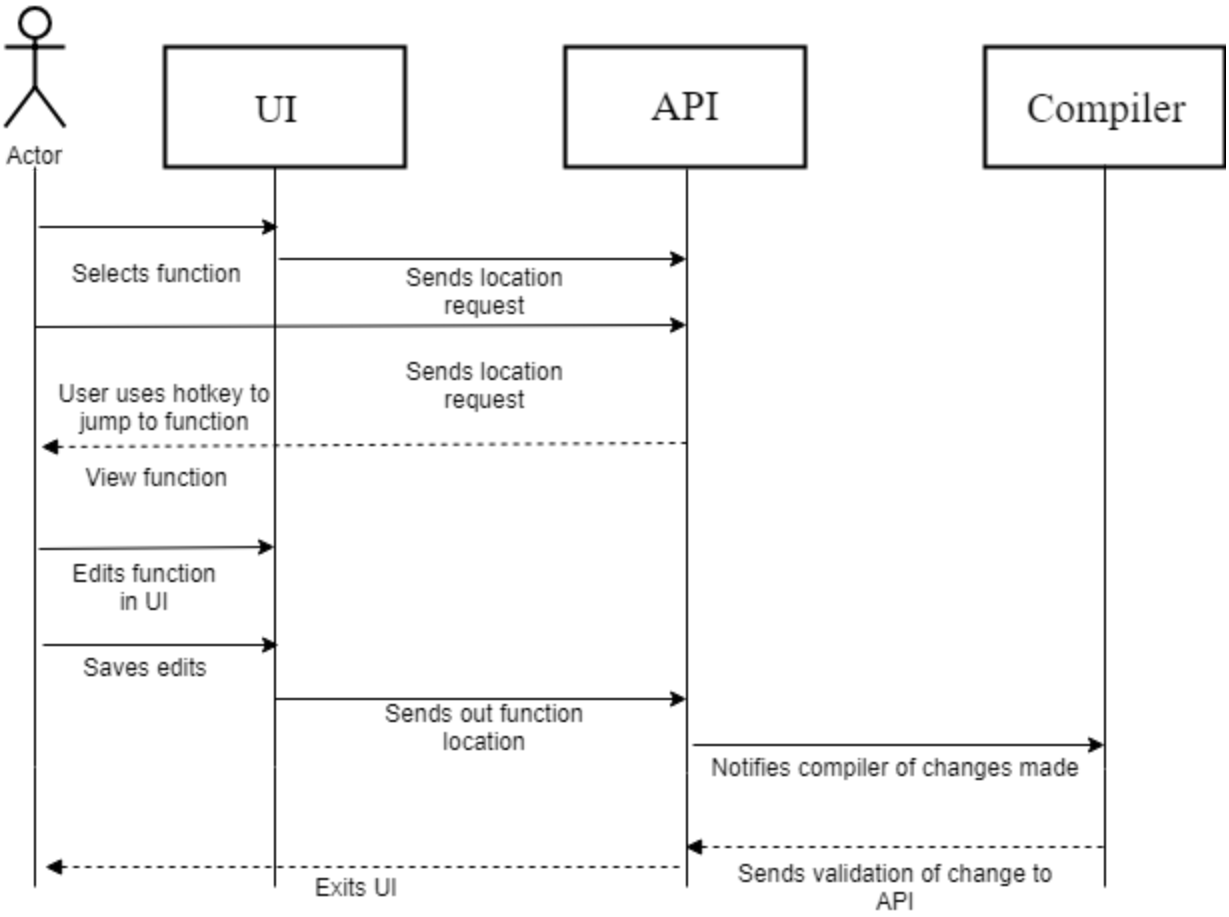
Peek Function



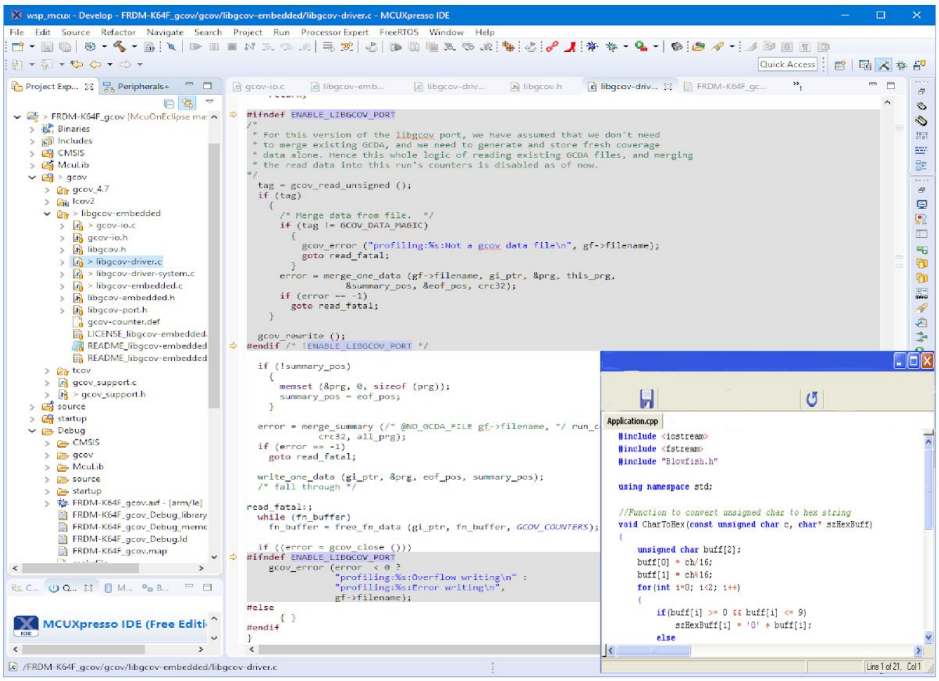
Jump to Function



Edit Function



# User Interface Design Sketches



# Glossary of Terms

**Integrated Development Environments (IDE):** An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools, and a debugger.

**Eclipse:** Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE.[6] It contains a base workspace and an extensible plug-in system for customizing the environment.

**Plug-in:** In computing, a plug-in (or plugin, add-in, addin, add-on, or addon) is a software component that adds a specific feature to an existing computer program.

**Open Source:** Open-source software (OSS) is a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose.

**Hotkey:** Keyboard shortcuts, or hotkeys,  are typically a means for invoking one or more commands using the keyboard that would otherwise be accessible only through a menu, a pointing device, different levels of a user interface, or via a command-line interface.

# References

Eclipse (software). (2018, December 28). Retrieved from https://en.wikipedia.org/wiki/Eclipse_(software)

Integrated development environment. (2019, February 13). Retrieved from https://en.wikipedia.org/wiki/Integrated_development_environment

Keyboard shortcut. (2019, January 24). Retrieved from https://en.wikipedia.org/wiki/Keyboard_shortcut

Open-source software. (2019, February 08). Retrieved from https://en.wikipedia.org/wiki/Open-source_software

Plug-in (computing). (2019, February 09). Retrieved from https://en.wikipedia.org/wiki/Plug-in_(computing)

*Welcome to Eclipse*[PDF]. (2005, February 24). IBM Corporation and others.