HTML EDITOR DOCS

Code editor for creating/editing HTML documents. The editor will have built-in functionalities that help the user when editing the document, such as: warnings about the correctness of the content, autocomplete, syntax highlighting, code refactoring, multiple cursors, etc. The editor will also have several options for setting up the environment.

UML diagrams

Use cases:

- Code editing
- Search in file
- Add new tab
- File tree view manipulation
- Editor settings

Actors:

- User

Use case diagram:

Code editing

Use case name:

Short description: The user opens the application and blanc document is opened. The user can edit the blanc document, crate a new document, open an existing document or save the document that is currently opened.

Actors:

- User: Edits HTML documents.

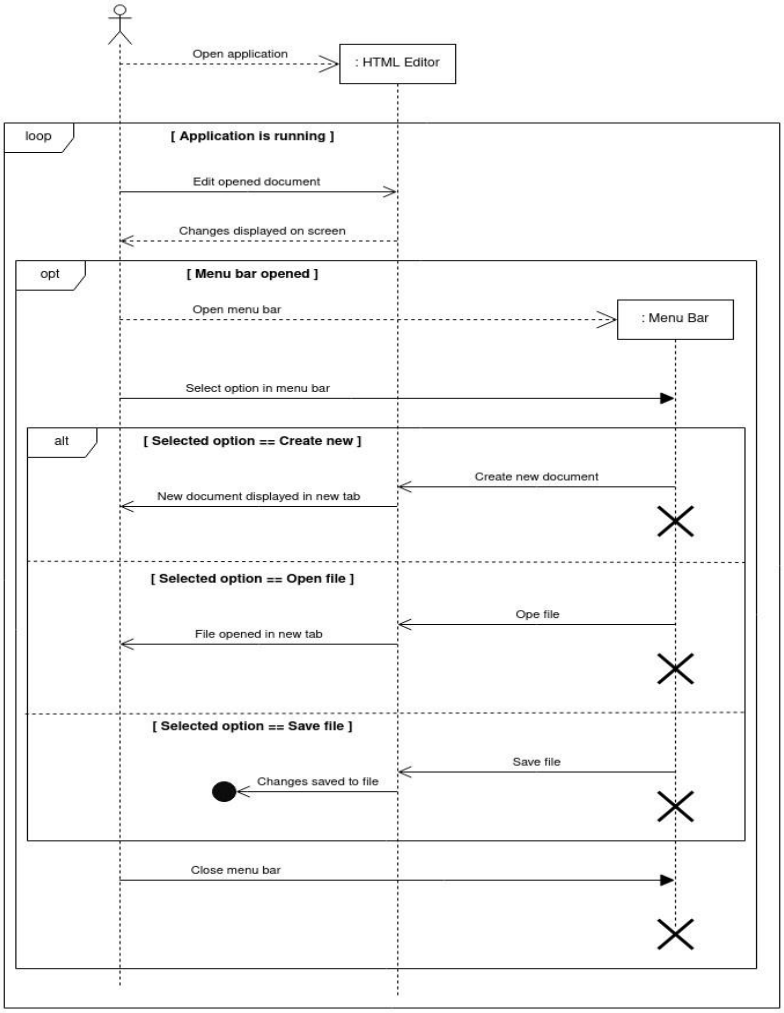
Preconditions: The application is opened and a new empty document is created.

Postconditions: The use can continue editing the document or can proceed to any other use case.

Main path:

1. The user opens the application.
2. A empty document is created in memory and is opened in the editor.
3. The user edits the content of the open documents. 3.1. The user can open the menu bar and do one of the following actions: 3.1.1. Create a new file. A new empty document is created in memory and opened in a new tab in the editor. Go back to step 3. 3.1.2. Open an existing file. 3.1.2.1. A file selection windows is opened. The user selects a file form the file system. 3.1.2.2. A document with the contents of the selected file is opened in a new tab in the editor. Go back to step 3. 3.1.3. Save to the file system all changes made to a document opened in memory. Go back to step 3.

Sequence diagram:



Use case name:

Search in file

Short description: The User can click on a *Search* button to search the contents of a file.

Actors:

- The User: clicks on a *Search* button.

Precondition: Code Editing use case.

Postcondition: The Code Editor displays all of the words that match the search.

Main path:

1. The User clicks on a *Search* icon.
2. The application opens a search prompt.
3. Until the User presses the `enter` key or `escape` key:
    1. The User types a word they want to search for.
4. If the User pressed the `enter` key:
    1. Application highlights all of the words that match the search.
5. Otherwise if the User pressed the `escape` key:
    1. The search is canceled and the prompt is removed.

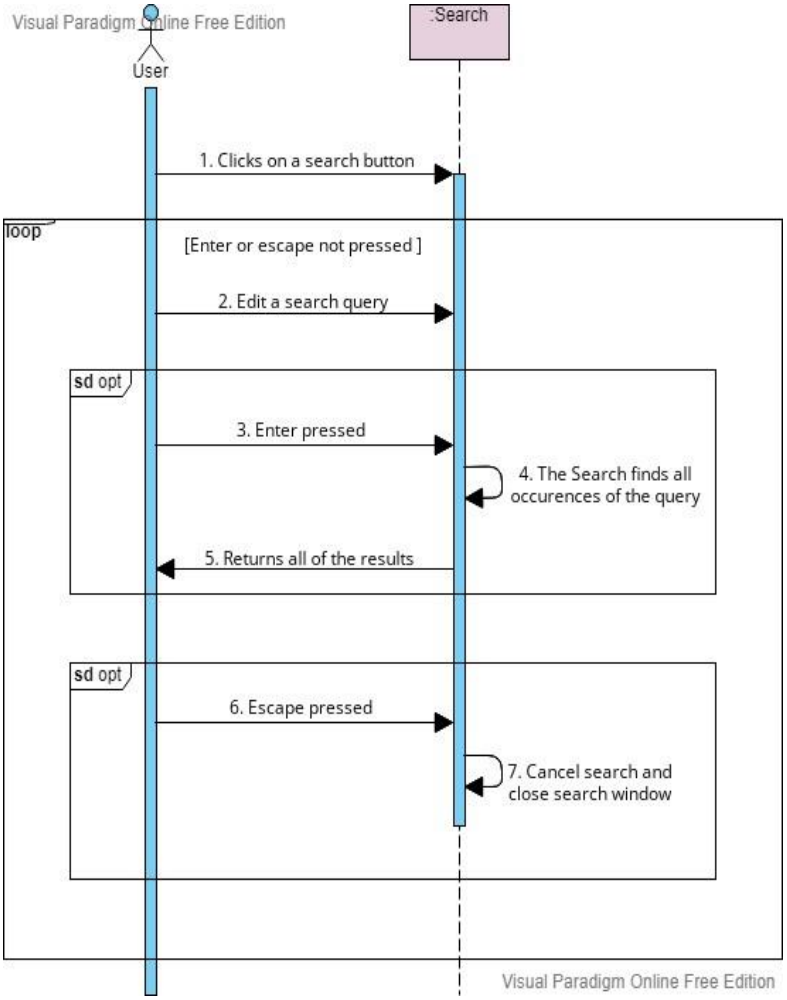
Alternative path:

- A1: Unexpected application exit: The search is discarded and the applications closes.

Additional information: No additional information.

Sequence diagram:

Use case name:



Add new tab

Short description: The user can add new tabs by opening existing documents or creating new ones, or remove open ones from tab navbar by clicking on the "x" symbol. They can switch between open tabs by using mouse clicks or ctrl+tab shortcut.

Actors: *User: clicks on certain file on the top of the editor (tab navbar).

Preconditions: The application is opened and a new empty document is created.

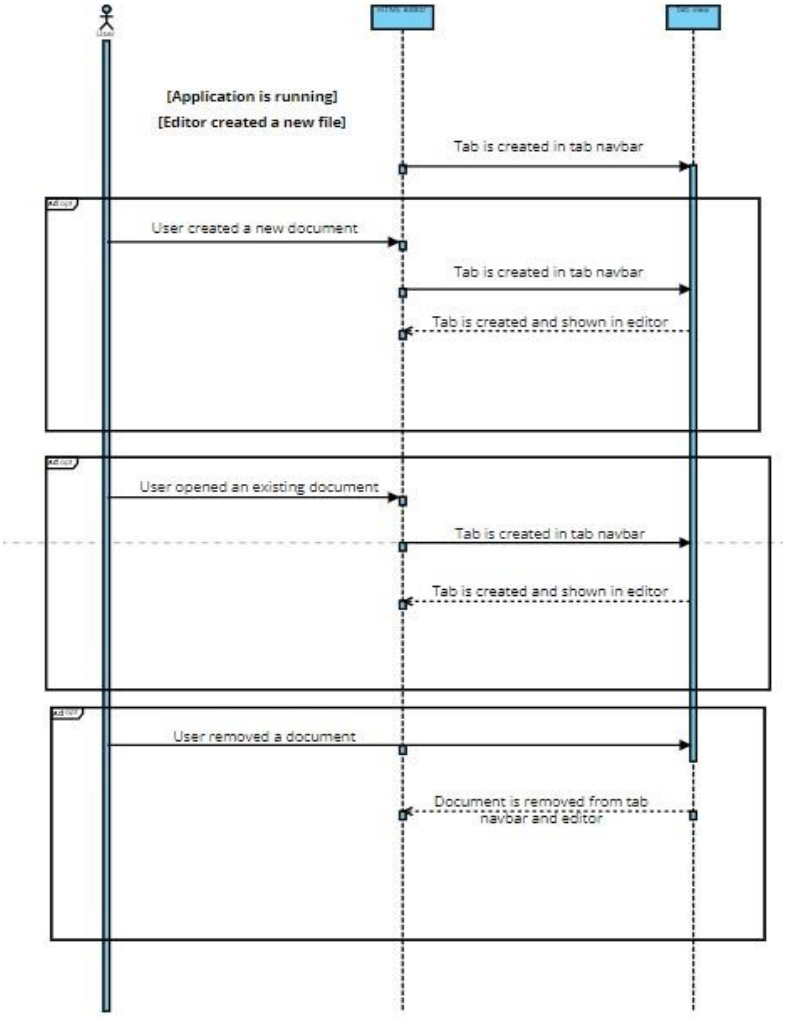
Postconditions: Working directory is changed based on users actions. User can procede to any other user case.

Main path:

1. The user opens the application.
2. An empty document is created and opened in the editor - tab that is representing the open document is visible on the top of the editor.
3. The user edits the content of the tab navbar 3.1 The user creates a new document A new empty document is created and it is opened in a new tab in the editor. Go back to step 3. 3.2 The user opens an existing document An existing document is opened and it is added in a new tab in the editor. Go back to step 3. 3.3 The user closes an opened tab 3.3.1 The document is not saved The editor will ask the user if they want to keep the changes they made to the document, and after saving or discarding the changes it will remove the document from the list of open documents. 3.3.2 The document is saved The document will be removed from the list of open documents.

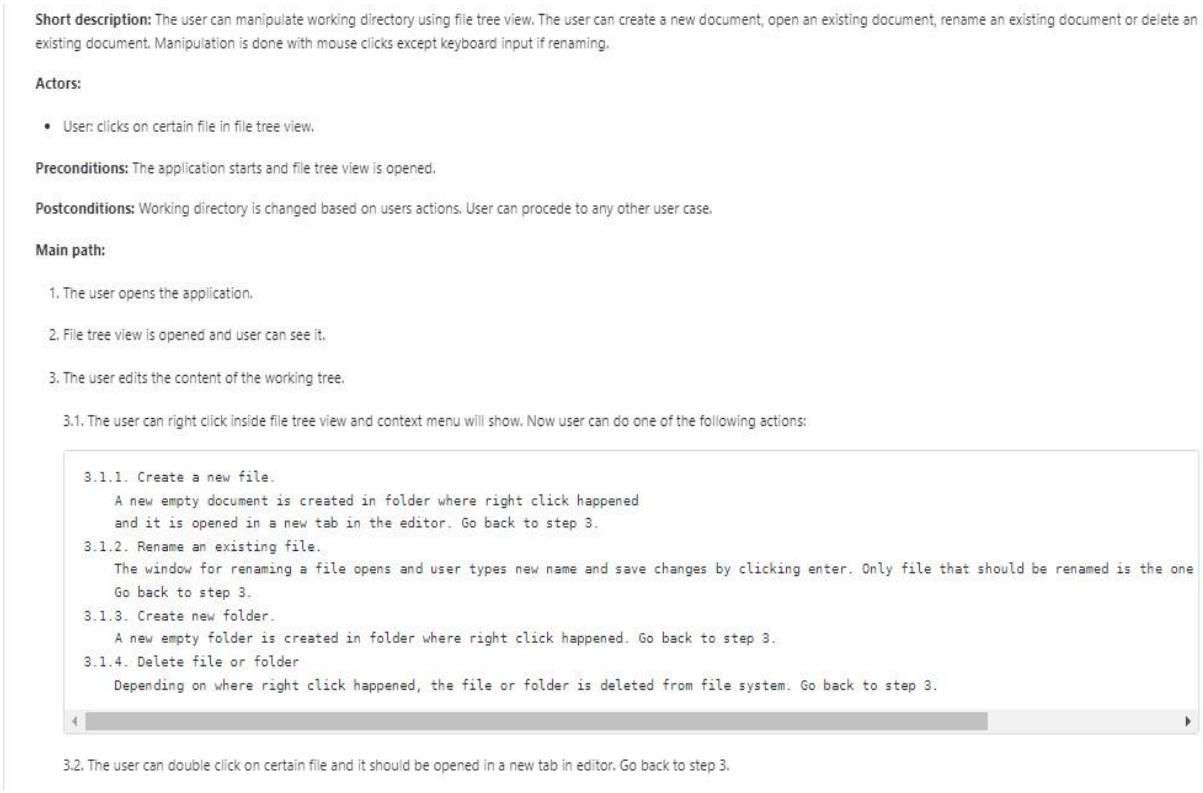
Sequence diagram:

Use case name:



File tree view manipulation

Use case name:



Short description: The user can manipulate working directory using file tree view. The user can create a new document, open an existing document, rename an existing document or delete an existing document. Manipulation is done with mouse clicks except keyboard input if renaming.

Actors:

- User: clicks on certain file in file tree view.

Preconditions: The application starts and file tree view is opened.

Postconditions: Working directory is changed based on users actions. User can procede to any other user case.

Main path:

1. The user opens the application.
2. File tree view is opened and user can see it.
3. The user edits the content of the working tree.

3.1. The user can right click inside file tree view and context menu will show. Now user can do one of the following actions:

```
3.1.1. Create a new file.
    A new empty document is created in folder where right click happened
    and it is opened in a new tab in the editor. Go back to step 3.
3.1.2. Rename an existing file.
    The window for renaming a file opens and user types new name and save changes by clicking enter. Only file that should be renamed is the one
    Go back to step 3.
3.1.3. Create new folder.
    A new empty folder is created in folder where right click happened. Go back to step 3.
3.1.4. Delete file or folder
    Depending on where right click happened, the file or folder is deleted from file system. Go back to step 3.
```

3.2. The user can double click on certain file and it should be opened in a new tab in editor. Go back to step 3.

*3.1.1. Create a new file.

A new empty document is created in folder where right click happened
and it is opened in a new tab in the editor. Go back to step 3.

3.1.2. Rename an existing file.

The window for renaming a file opens and user types new name and save changes by clicking enter. Only file that should be renamed is the one on which user right clicked.

Go back to step 3.

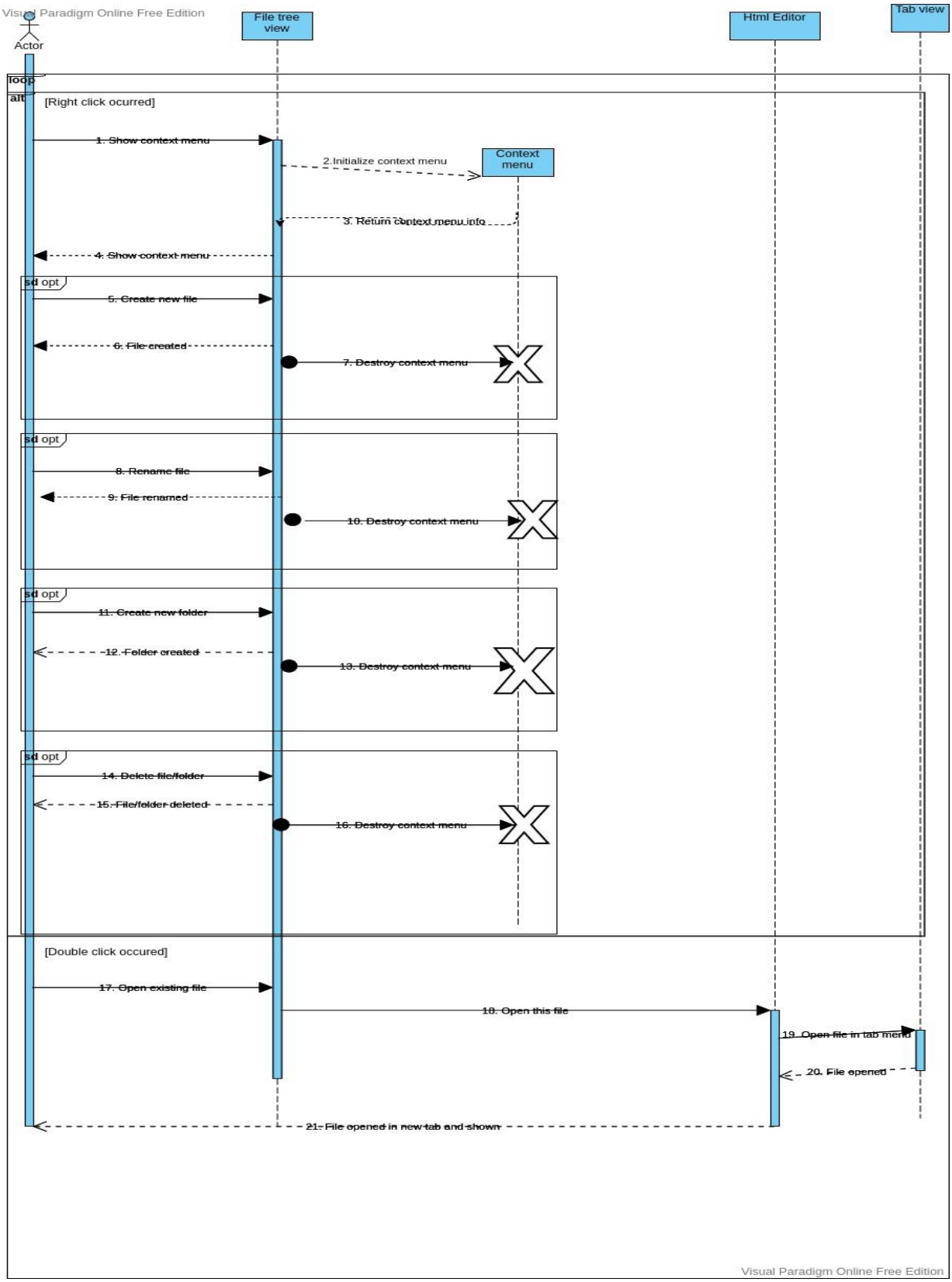
3.1.3. Create new folder.

A new empty folder is created in folder where right click happened. Go back to step 3.

3.1.4. Delete file or folder

Depending on where right click happened, the file or folder is deleted from file system. Go back to step 3.

Sequence diagram:



Use case name: Editor settings

Short description: The user can set editor properties like font size, editor theme or view options

Actor: User: clicks on desired property and select offered values.

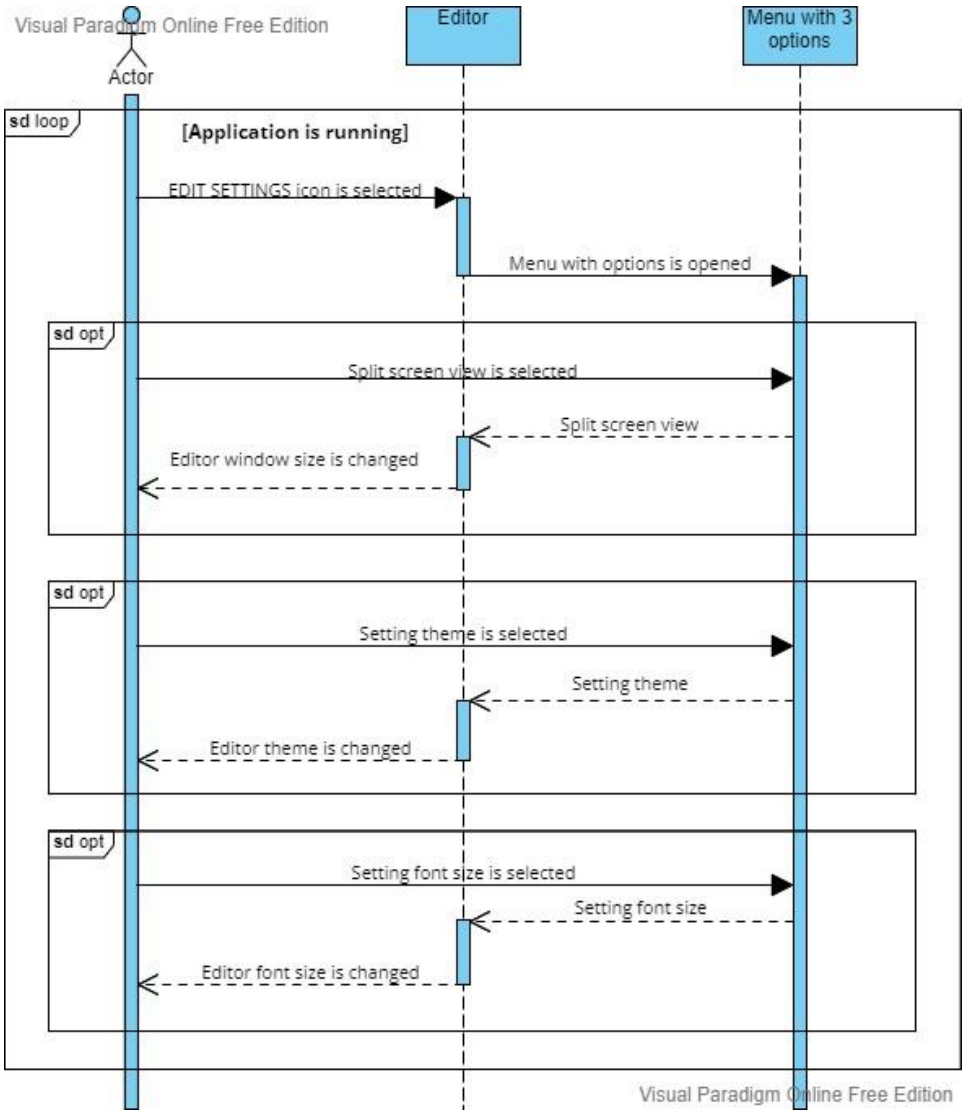
Preconditions: The application is running.

Postconditions: Editor properties are changed to selected values.

Main path:

1. Application is running (blank document or editing mod)
2. User clicks on EDITOR SETTINGS icon from Menu
3. Menu with three choices is opened. 3.1 Split screen view. Editor window size is changed to fit split screen mod. Go back to step 1. 3.2. Setting theme 3.2.1. Window with offered values is opened. 3.2.2. If one of the themes is selected, editor property is changed. Go back to step 1. 3.3. Setting font size. 3.3.1. Window with offered values and input label is opened. 3.3.2. If one of the themes is selected or entered, editor property is changed. Go back to step 1.

Sequence diagram:



Class diagram