

EX.NO:4

DATE :

TEXT EDITOR

AIM

To prepare necessary documents and to develop the “Text Editor” with usecase diagrams using software engineering methodology.

PROBLEM STATEMENT

The project Text Editor is to develop an application for allows the user to type the text with different font styles, type, size and vibrant colors. It contains facility to save the typed text as document and to open the existing document for reading and writing.

SOFTWARE REQUIREMENT ANALYSIS

1. Editor Form
2. Font Style Frame
3. Save Frame
4. Open Frame

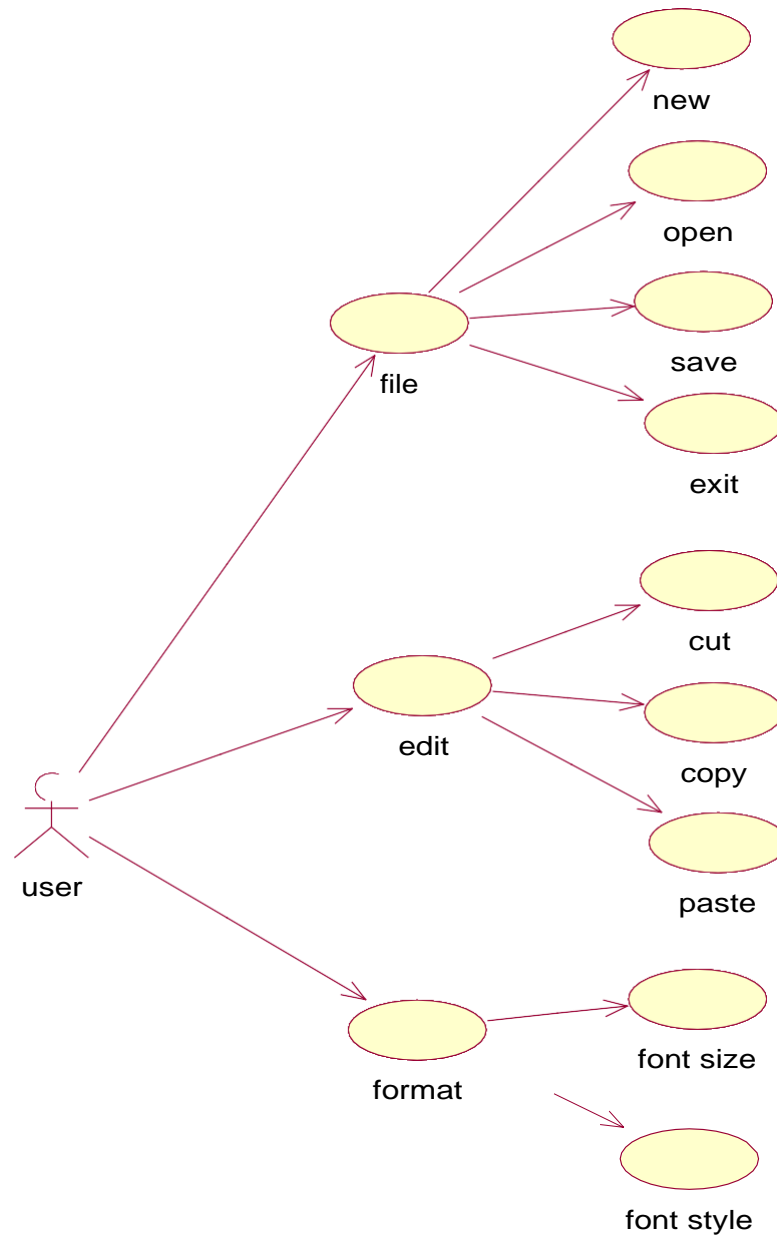
The Editor form contains the components that allow the user to type the text and the menu bar that provides various options.

The Font style, Save and Open frames are the members of the Editor form.

The Font Style frame allows the user to enrich their document with different font color, style, type and size of the text.

The Save frame allows the user to save their typed text as document and the Open frame allows the user to open the existing document for reading and writing.

USE CASE DIAGRAM



Use case Diagram:

Use case diagram is a graph of actors, set of use cases enclosed by a system boundary, communication (participation) association between the actors and the use cases and a generalization among the use cases.

Use case:

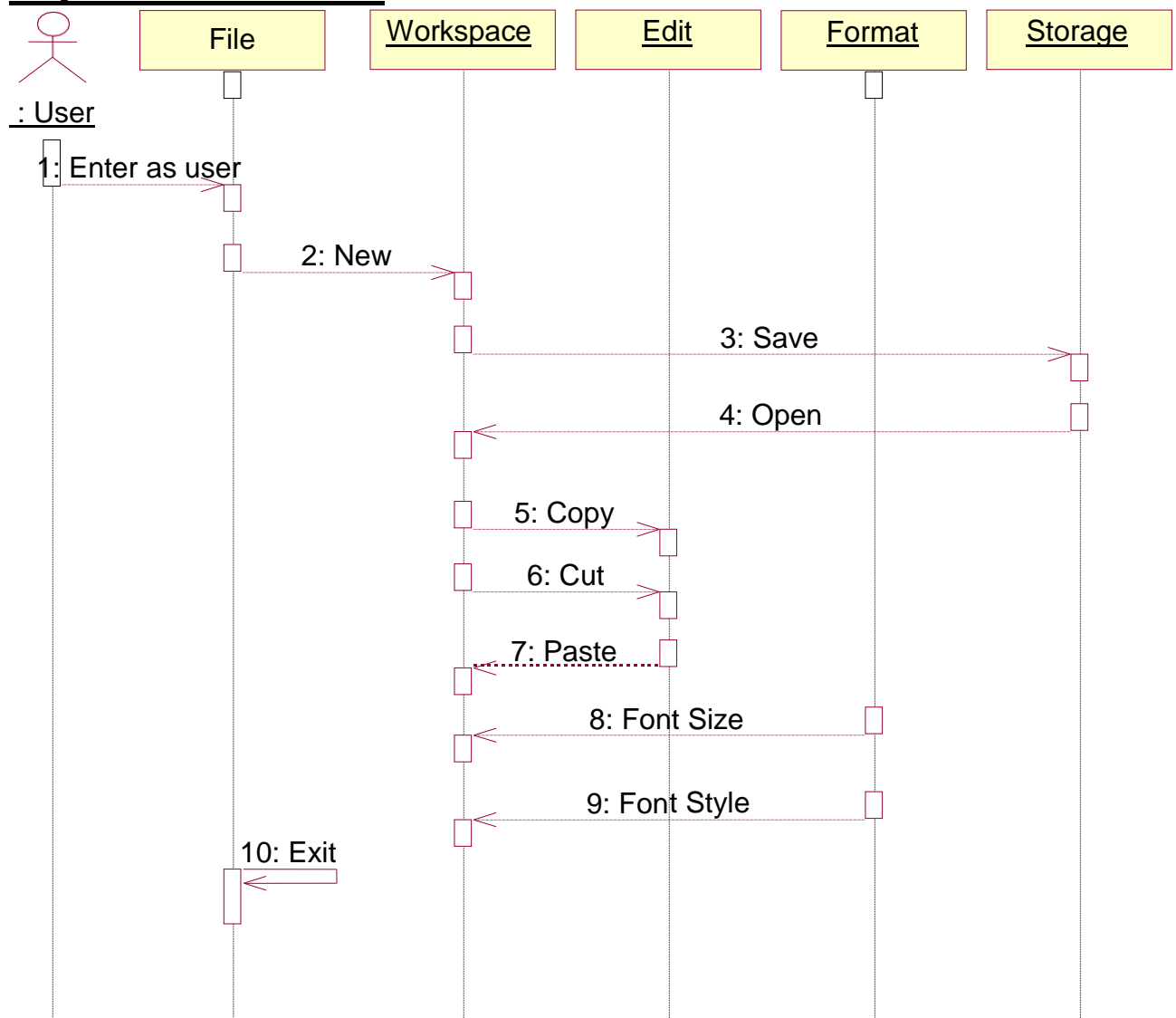
A use case is a description of a set of sequence of actions that a system performs to yield result of value to an actor.

The Use Cases described are,

1. New File
2. Type Text
3. Format Text
4. Change Style
5. Open, Save File

The File use case is to describe that, the user should Open and Save the files or create a new File.

SEQUENCE DIAGRAM:



Sequence diagrams are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence.

The objects used in this sequence diagram are,

1. File
2. Workspace
3. Edit
4. Format
5. Storage

BASIC FLOW:

The user opens the application and application allows the user to type the text.

ALTERNATE FLOW:

The object File makes the user to select new file or to open the existing file and to save the file.

The object Workspace allows the users to type text, from the workspace the users access all other object.

The object Edit used to cut, copy and paste the selected text.

The object Format allows the user to change the font style, color and its type.

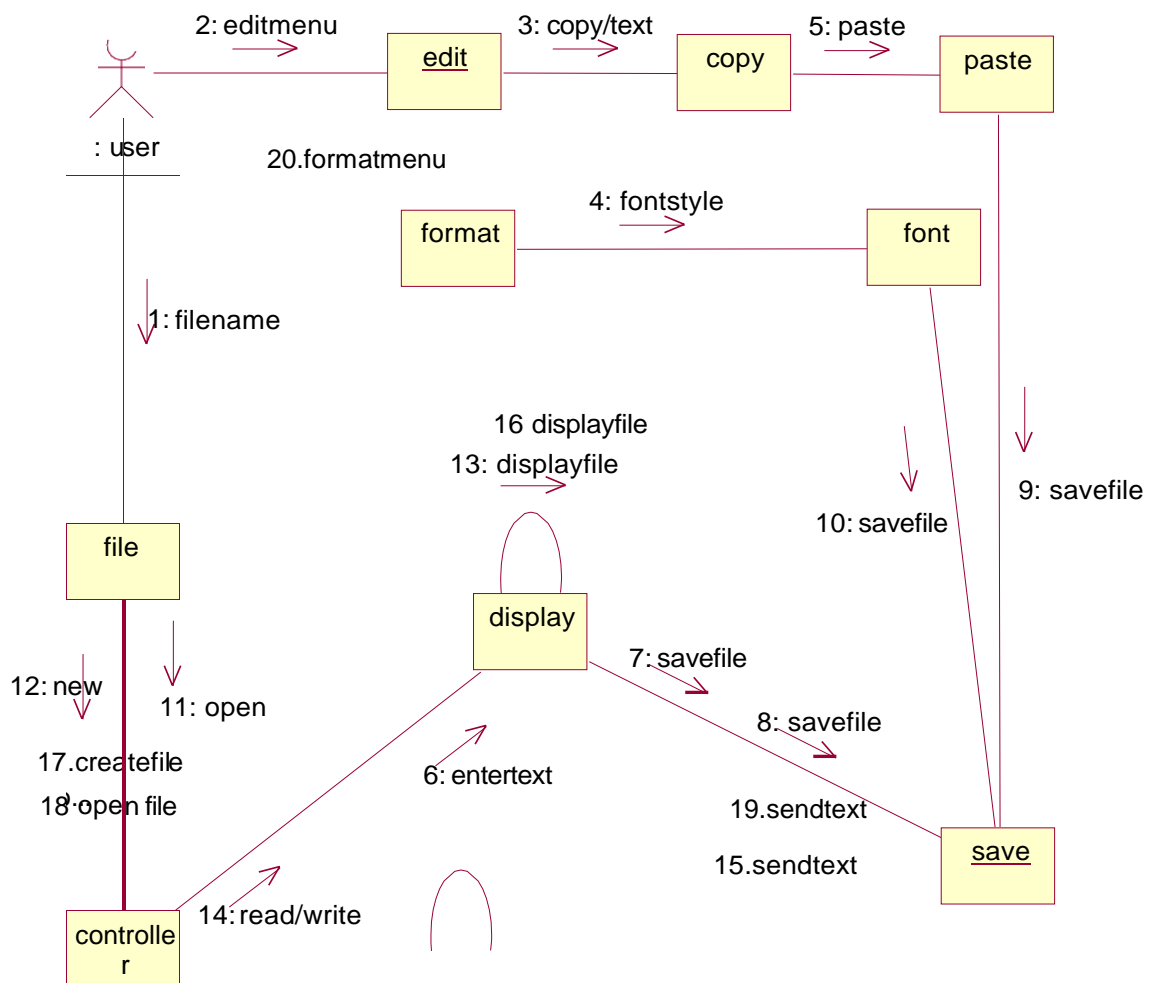
The object Storage is to store the typed text in the disk as file.

COLLABORATION DIAGRAM

A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome.

Collaboration diagram shows exactly the same information as the sequence diagram. However, collaboration diagram shows this information in a different way and with different purpose.

In this collaboration diagram, the objects are represented as rectangle, the actors are stick figures. Whereas the sequence diagram illustrates the object and actor interaction overtime, the collaboration diagram shows the object and actor interaction without reference to time.



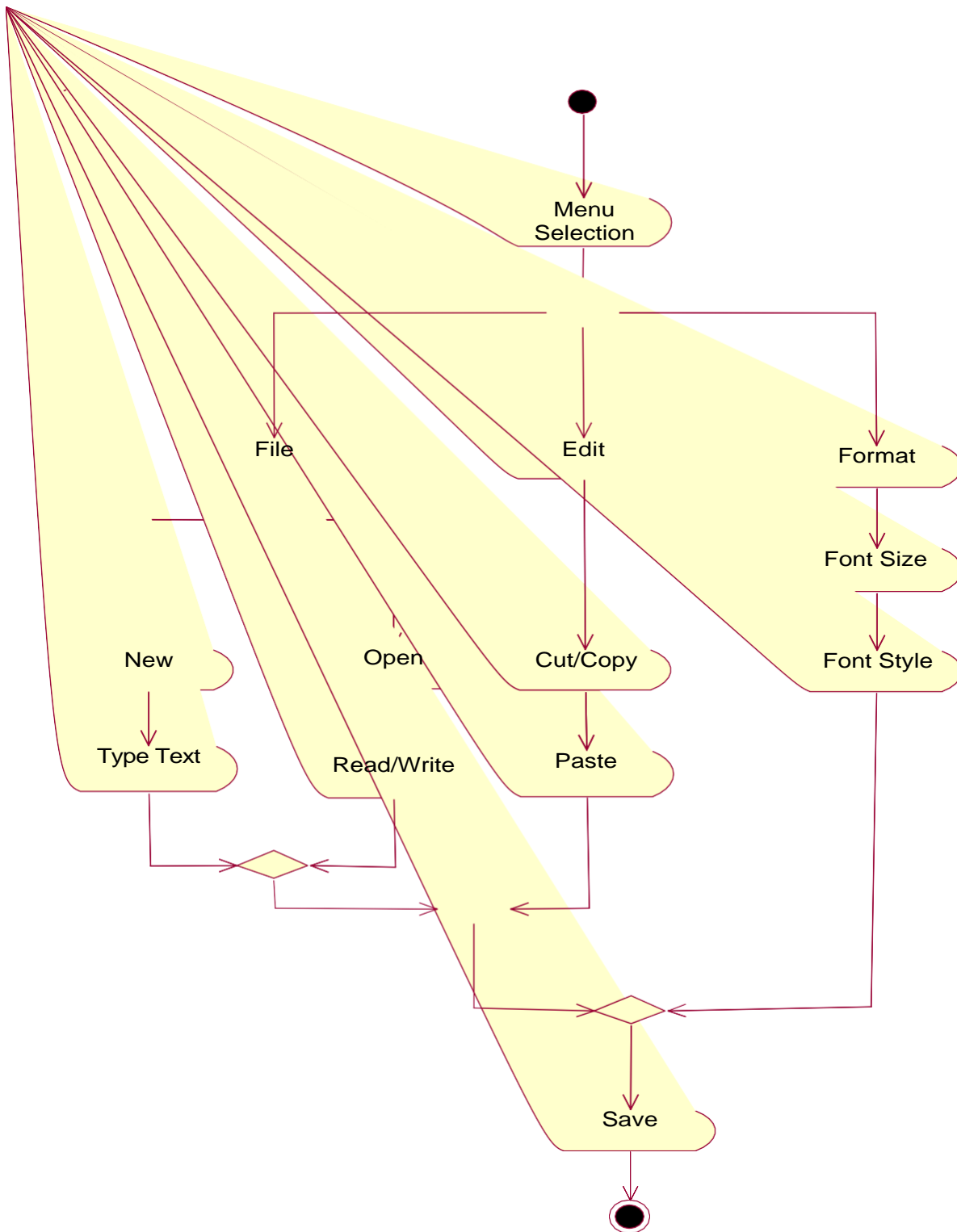
In our TEXT EDITOR each object interacts with each other or collaborates with each other; it gets represented by the solid line drawn between them.

ACTIVITY DIAGRAM

The activity diagram describes the sequencing of activities with support for both conditional and parallel behavior.

The Activity diagram is used to describe the various activities taking place in an application. Here in our TEXT EDITOR, we have various activities starting from File.

After user opens the application he/she choose file menu to create new file or to open the existing file then user read/write. User chooses the Edit menu to cut, copy and paste the selected text.

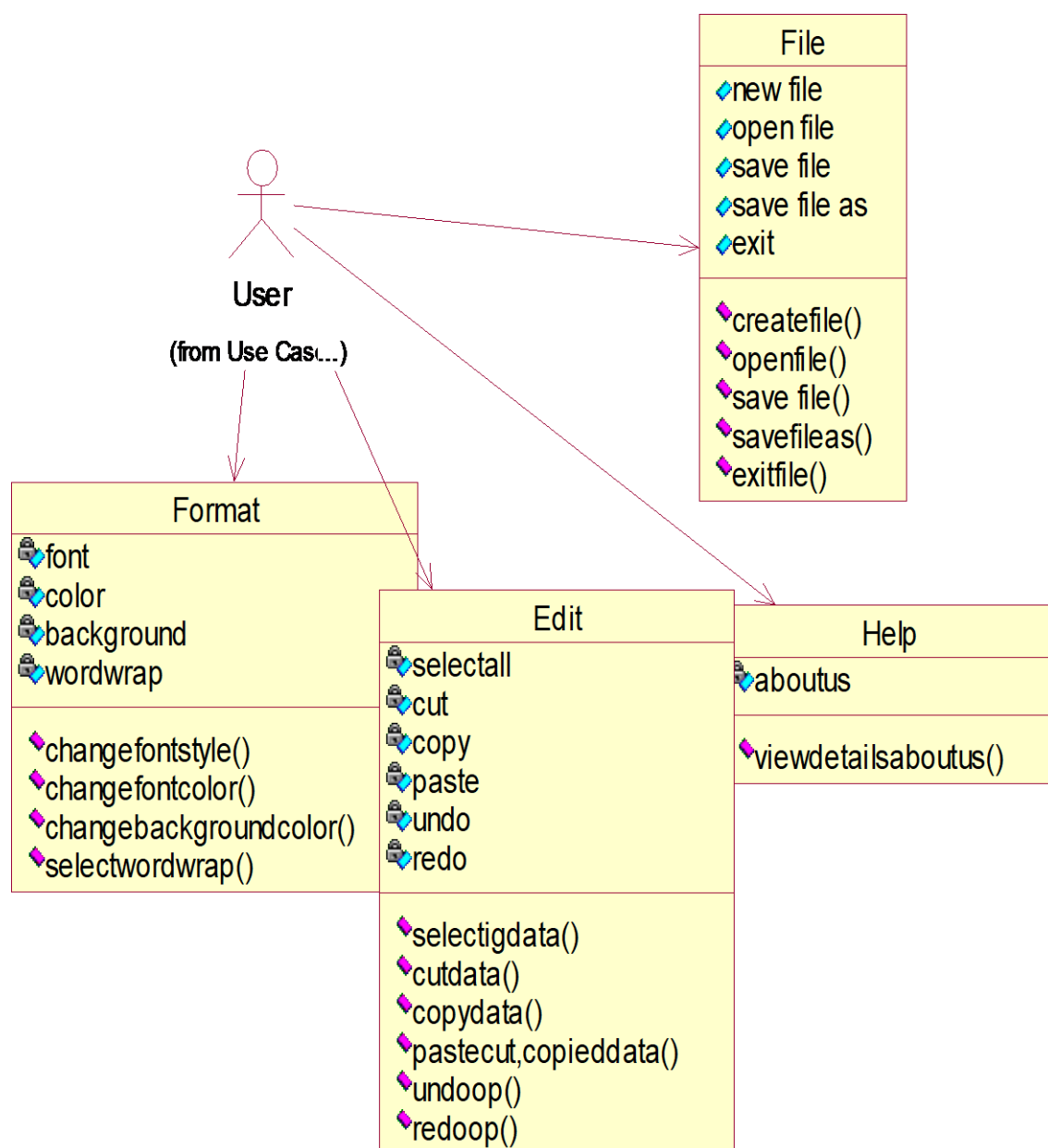


By choosing Format the user can change the font style, color and its type. The object Storage is to store the typed text in the disk as file.

CLASS DIAGRAM:

Class diagrams show the interactions between classes in the system. Class diagram also shows the attributes and operation of a class and the constraints that apply to the way objects are connected.

Classes contain information and behavior that acts on that information. Each class on class diagram is represented by rectangle divided into three sections. The first section shows the class name, second section shows the attributes the class contains and last section contains the operation of the class.



SOFTWARE DEVELOPMENT

Fore color:

```
Private Sub mnuTxtColor_Click()  
cd.ShowColor  
FrmChild.TextBox.ForeColor = cd.Color  
End Sub
```

Background Color:

```
Private Sub mnuBackColor_Click()  
cd.ShowColor  
FrmChild.TextBox.BackColor = cd.Color  
End Sub
```

Fonts:

```
Private Sub mnuFont_Click()  
cd.Flags = cd1CFBoth  
cd.ShowFont  
FrmChild.TextBox.Font = cd.FontName  
FrmChild.TextBox.FontItalic = cd.FontItalic  
FrmChild.TextBox.FontBold = cd.FontBold  
FrmChild.TextBox.FontSize = cd.FontSize  
End Sub
```

Exit:

```
Private Sub mnuExit_Click()  
End  
End Sub
```

SOFTWARE TESTING:

TEST CASES

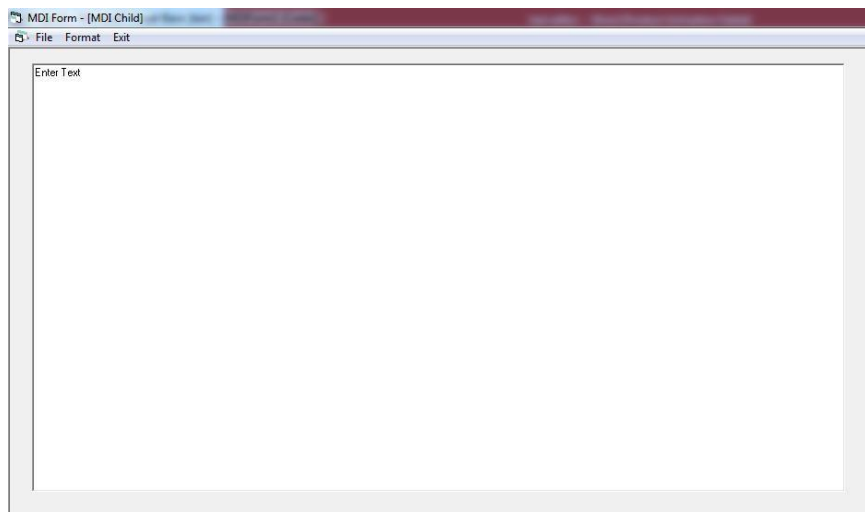
The sample text is typed and its font style, font color and font type are changed.

The new file is created and the typed text is stored in disk as file. The existing file is opened and it allows to read or write the file content.

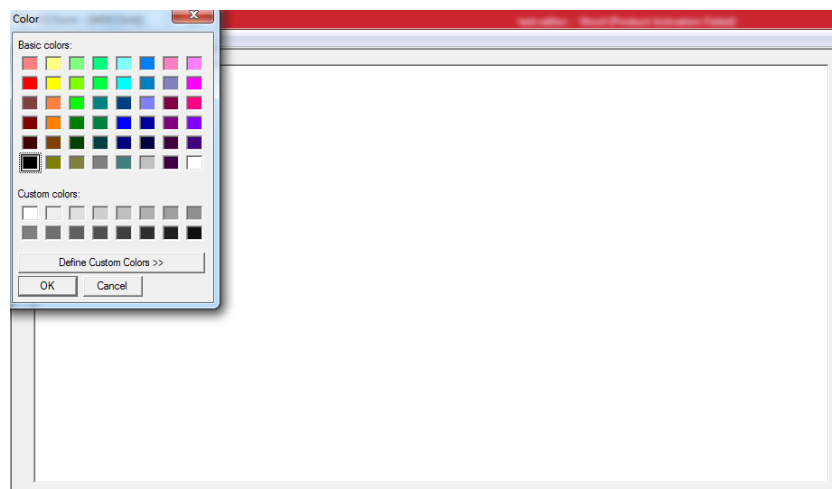
The developed application gets tested with the sample inputs.

OUTPUT:

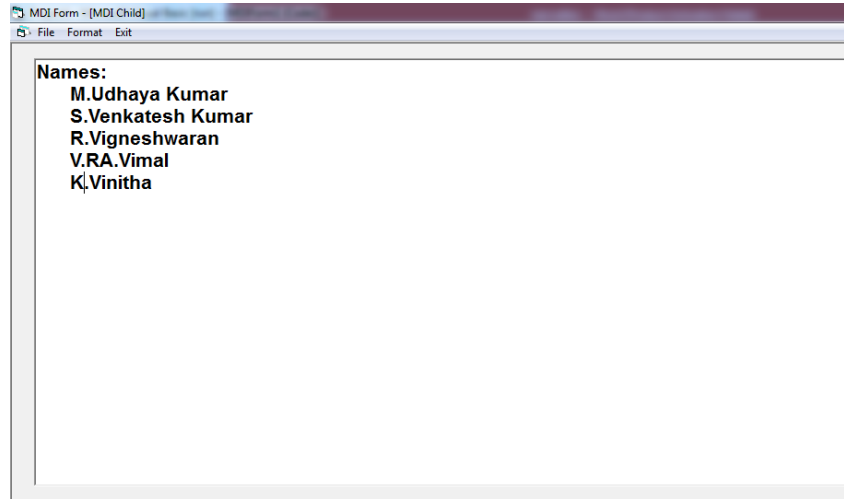
Form1:



Form2:



Form3:



RESULT:

Thus the Text Editor is developed with all necessary documents and usecase diagrams using software engineering methodology.