##### D-PAD TEXT EDITOR

**A PROJECT REPORT**

###### ***Submitted by***

##### Kanika(19BCE10165)

##### Rishita Srivastava(19BCE10121)

##### Soumya Singh(19BCE10159)

##### Yah Thakkar(19BCE10388)

*in partial fulfillment for the award of the degree*

*of*

##### BACHELOR OF TECHNOLOGY

*in*

**COMPUTER SCIENCE ENGINEERING(CSE CORE)**



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH - 466114**

APRIL 2021

**VIT BHOPAL UNIVERSITY,KOTHRIKALAN, SEHORE**

**MADHYA PRADESH – 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**D-PAD**TEXT EDITOR”is the bonafide work of **Kanika (19BCE10165) , Rishita Srivastava (19BCE10121) , Soumya Singh (19BCE10159) , Yash Thakkar (19BCE10388)** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project / research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR PROJECT GUIDE**

Sandip Mal, Assistant ProfessorAssistant Professor Nidhi Mishra, Senior Assistant Professor

School of Computer Science and Engineering School of Computer Science and Engineering

VIT BHOPAL UNIVERSITY VIT BHOPAL UNIVERSITY

The Project Exhibition II Examination is held on 29-04-2021.

**ACKNOWLEDGEMENT**

First and foremost I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr. Manas Kumar Mishra Head of the Department, School of Computer Science Engineering for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide Ms. Nidhi Mishra for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computer Science Engineering, who extended directly or indirectly all support.

Last, but not the least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.**  **1.1**  **1.2** | **TITLE**  System architecture diagram  Module workflow diagram | **PAGE NO.**  17  24 |

**ABSTRACT**

A text editor is a program that allows the user to open, view, and edit plain text files (files containing only text). They are essential to the world today.

Text editors deal with manipulating text and provide features to enhance the writing experience. The major functionalities of text editors are: inserting, deleting, and viewing text. Additional features are: find and replace, copy/cut and paste, text formatting, sentence highlighting, and etc.

With their relevance in every programming, sub-culture Text Editors tops the list since text-editor is the basic tool everyone uses.

The bare backbones of a text editor heavily relies on the data structures. Data Structures are used to add major functionalities and features in a text editor. There are a lot of text editors available. There are those that run in the terminal in GUI in browsers and in Browser engines. Many are very good and some are great. But sometimes, the most satisfying answer to any question is the one you build yourself.

So we have decided to create a unique Text Editor **D-Pad**, by using various data structures and java programming where users will not only be able to pen down their ideas thoughts notes To-Do lists but along with that the users will be able to express their emotions while writing and add spice to their text by using a doodling palette where either they can create their own doodles using the draw feature where the user will be able to choose different brushes and colors and draw different doodles and graphs as required or choose from the available doodles of different categories enhancing the overall text and writing experience.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | List of Figures  Abstract | iv  v |
| 1 | **INTRODUCTION** Introduction 1.2 Motivation for the work  1.3 Problem Statement  1.4 Objective of the work | 1-2  1  2  2  2 |
| 2 | **LITERATURE REVIEW**  2.1 Introduction  2.2 Existing Algorithms  2.3 Summary  ***Note: This should be written as per Ph.D thesis format of HITS.(Specimen attached)*** | 3 |
| 3 | **SYSTEM ANALYSIS**  3.1 Introduction  3.2 Disadvantages/Limitations in the existing system  3.3 Proposed System and methodology  3.4 Module Description  3.4.1 File Module  3.4.2 Edit Module  3.4.3 Format Module  3.4.4 View Module  3.4.5 Sketching Doodles | 4-10  4  4  5  6-10 |
| 4 | **SYSTEM DESIGN AND IMPLEMENTATION**  4.1 Introduction  4.2 System Architecture Diagram  4.3 Hardware and software requirements  4.4 Module Implementation WorkFlow  4.5 Module WorkFlow Diagram | 11-15  11  12  13  14-15 |
|  |  |  |
| 5 | **FUTURE ENHANCEMENT AND CONCLUSION**  6.1Real Time Usage  6.2 Novelty  6.3 Future Scope  6.4 Results and Conclusion | 16-18  16  17  18 |
| 6 | **REFERENCES**  ***Note: List of References should be written as per IEEE/Springer reference format. Arranged in alphabetical order. (Specimen attached)*** | 19 |

**INTRODUCTION**

A text editor is a program that allows the user to open, view, and edit plain text files (files containing only text). They are essential to the world today.

Text editors deal with manipulating text and provide features to enhance the writing experience. The major functionalities of text editors are: inserting, deleting, and viewing text. Additional features are: find and replace, copy/cut and paste, text formatting, sentence highlighting, and etc.

With their relevance in every programming, sub-culture Text Editors tops the list since text-editor is the basic tool everyone uses.

The bare backbones of a text editor heavily relies on the data structures. Data Structures are used to add major functionalities and features in a text editor. There are a lot of text editors available. There are those that run in the terminal in GUI in browsers and in Browser engines. Many are very good and some are great. But sometimes, the most satisfying answer to any question is the one you build yourself.

So we have decided to create a unique Text Editor **D-Pad**, by using various data structures and java programming where users will not only be able to pen down their ideas thoughts notes To-Do lists but along with that the users will be able to express their emotions while writing and add spice to their text by using a doodling palette where either they can create their own doodles using the draw feature where the user will be able to choose different brushes and colors and draw different doodles and graphs as required or choose from the available doodles of different categories enhancing the overall text and writing experience.

**1**

**MOTIVATION FOR THE WORK**

There are a lot of text editors available.  with  functionalities like **inserting, deleting, and viewing text** and features like **find and replace, copy/cut and paste, text formatting, sentence highlighting, and etc**.

Many are very good and some are great. But sometimes, the most satisfying answer to any question is the one you build yourself.

Welcome to 2021, where doodles are among the most important  tool while writing ,  But the existing text editors are only focussed only towards editing texts and does not include doodling features and draw options which becomes a major drawback of the existing text editors as people today urge and look for doodles graphical representations like graphs to express their thoughts while writing and present it in a better way.

**PROBLEM STATEMENT**

To create a unique text editor that not only edits and formats the text but also have some unique features to enhance the overall writing experience of the user using data structures and java programming.

**OBJECTIVE**

To create a unique text editor using data structures and java programming that not only edits and formats the text but also have some unique features such as doodling inserting images etc. to enhance the overall writing experience.

**2**

**LITERATURE REVIEW**

A **text editor** is a type of [computer program](https://en.wikipedia.org/wiki/Computer_program) that edits [plain text](https://en.wikipedia.org/wiki/Plain_text). Such programs are sometimes known as "**notepad**" software, following the naming of [Microsoft Notepad](https://en.wikipedia.org/wiki/Microsoft_Notepad). Text editors are provided with [operating systems](https://en.wikipedia.org/wiki/Operating_system) and software development packages, and can be used to change files such as [configuration files](https://en.wikipedia.org/wiki/Configuration_file), documentation files and [programming language](https://en.wikipedia.org/wiki/Programming_language) [source code](https://en.wikipedia.org/wiki/Source_code).

Before text editors existed, computer text was punched into [cards](https://en.wikipedia.org/wiki/Punched_cards) with [keypunch](https://en.wikipedia.org/wiki/Keypunch) machines. Physical boxes of these thin cardboard cards were then inserted into a card-reader. Magnetic tape and disk "card-image" files created from such card decks often had no line-separation characters at all, and assumed fixed-length 80-character records. An alternative to cards was punched paper tape. It could be created by some [teleprinters](https://en.wikipedia.org/wiki/Teleprinter) (such as the Teletype), which used special characters to indicate ends of records.

The first text editors were ["line editors"](https://en.wikipedia.org/wiki/Line_editor) oriented to teleprinter- or [typewriter](https://en.wikipedia.org/wiki/Typewriter)-style terminals without displays. Commands (often a single keystroke) effected edits to a file at an imaginary insertion point called the "cursor". Edits were verified by typing a command to print a small section of the file, and periodically by printing the entire file.When [computer terminals](https://en.wikipedia.org/wiki/Computer_terminal) with video screens became available, [screen-based text editors](https://en.wikipedia.org/wiki/Visual_editor) (sometimes called just "screen editors") became common. One of the earliest full-screen editors was [O26](https://en.wikipedia.org/wiki/O26_(text_editor)), which was written for the operator console of the [CDC 6000 series](https://en.wikipedia.org/wiki/CDC_6000_series) computers in 1967.

The core data structure in a text editor is the one that manages the string (sequence of characters) or list of [records](https://en.wikipedia.org/wiki/Storage_record) that represents the current state of the file being edited. While the former could be stored in a single long consecutive [array](https://en.wikipedia.org/wiki/Array_data_structure) of characters, the desire for text editors that could more quickly insert text, delete text, and undo/redo previous edits led to the development of more complicated sequence data structures. A typical text editor uses a [gap buffer](https://en.wikipedia.org/wiki/Gap_buffer), a [linked list](https://en.wikipedia.org/wiki/Linked_list) of lines (as in [PaperClip](https://en.wikipedia.org/wiki/PaperClip)), a [piece table](https://en.wikipedia.org/wiki/Piece_table), or a [rope](https://en.wikipedia.org/wiki/Rope_(data_structure)), as its sequence data structure.Two of the first text editors were Text Editor & Corrector (TECO) and ED, which were released in 1962 and 1969, respectively. As you can see on the computer digital representations of TECO below, the first text editors were not very complex.

However, they marked a big step in the progression towards modern text editors. As seen in the photo of a sample code in Sublime Text below, we have evolved a long way since the pioneer text editors.

**3**

**SYSTEM ANALYSIS**

**INTRODUCTION**

In this section we are going to first discuss the existing text editor and their limitations and then we’ll give a description of the proposed work and the various methodology used in the project. After describing the proposed work we’ll give a description of the 4 modules( file module, edit module,format module,sketching doodles) used in the project and discuss their intricate details.

**EXISTING WORK WITH LIMITATIONS**

There are a lot of text editors available.  with  functionalities like **inserting, deleting, and viewing text** and features like **find and replace, copy/cut and paste, text formatting, sentence highlighting, and etc**.

Many are very good and some are great. But sometimes, the most satisfying answer to any question is the one you build yourself.

Welcome to 2021, where doodles are among the most important  tool while writing ,  But the existing text editors are only focussed only towards editing texts and does not include doodling features and draw options which becomes a major drawback of the existing text editors as people today urge and look for doodles graphical representations like graphs to express their thoughts while writing and present it in a better way.

Your mundane texts should include a doodle to add a dash of color or a blip of personality. It is important to emphasize our texts with our body language and facial expressions, but technological advances make that a difficult task.

However, doodles make the task less daunting and more comprehensible.

**4**

**PROPOSED WORK AND METHODOLOGY**

D-PAD is a Text Editor which will include File operations, Data structures and will be created in Java.

Dpad will consist of File Handling operations to save , edit, print and delete text documents.

Text editing in Dpad will be achieved by using data structures like stack queue gap buffer in functions like cut, copy, paste, undo and redo.

Dpad  will have functions to change the Format of the text considering user preferences providing various  Font-families, font-style, font-color and font-size  accomplished by diverse inbuilt java libraries .

For user convenience and elevated options Dpad will include variety of view options allowing the user to set themes to the executable application, vary the size of the text in their document and for additional ease have a spell check functionality.

For Sketching , we’ll use the package org.eclipse.swt.graphics contains classes that allows management of graphics resources.

The subclass org.eclipse.swt.widgets. Canvas has a number of style bits that can be used to affect how painting occurs. We’ll be also using  presaved dataset to search for doodles by typing it’s category.

**5**

**MODULE DESCRIPTION**

D-PAD has been divided into 5 modules the 5 modules are as follows:

FILE MODULE

* The main typing area has been included using JTextArea
* File module consists of Open, Save , Print , Close.
* The open function usesJFileChooser from the java swing library. It is an effective way to prompt the user to choose a file along with this. We have also used JFileNameExtensionFilter which allows the user to choose and open files with only.txt extension.
* We use JFileChooser to save the file in the desired directory.
* The object of JFileChooser class represents a dialog window from which the user can select a file to either open a new file or save the created file in a specific directory.
* In this we create an empty constructor that points to user’s default directory
* **JFileChooser(String)** – uses the given path and then we locate our file through this path and open it.
* To ensure that only .txt files are used we have used JFileNameExtentionFilter this sets the file filter and allows us to add only .txt file and HTML files.
* At last we used JFrame.EXIT\_ON\_CLOSE to set the default close operation

6

EDIT MODULE:

* Copy and paste functionalities use data structures like gap buffer and rope data structure.
* The copy and paste commands are done using system calls. When copying or pasting the location of the file is saved onto  “clipboard” that can hold small amount of data for a small period of time.
* Example: Consider an example with initial gap size 10, initially, array or gap are of the same size, as we insert the elements in the array similarly elements will be inserted in the gap buffer, the only difference is gap size reduces on each insert.
* Basic operations in Gap Buffer
* Insert:It first checks whether the gap is empty or not, if it finds that the gap is empty it calls procedure grow() and resizes the gap and now the element can be inserted.
* Left : It is a procedure used for moving the cursor to the left and this cursor point is used as position for changes.
* Right : It is a procedure used for moving the cursor to right and this cursor point is used as a position for changes.
* Grow : It is a procedure used when the gap size has become zero and therefore we need to resize the array by inserting a gap at desired position.
* undo and redo functionalities are made using stack.
* For undo and redo functions we have used the UndoManager Library.
* Undo: Erases the last change made to the document
* Redo: Restores the most recent UNDO operations performed on the document.
* There are 2 stacks created , one each for undo and redo . Undo stack has a certain size in which last WRITE OPERATIONS encountered are stored.
* When undo is encountered the last operation is popped from the undo stack and added to the redo stack.
* If the user does a redo operation the top most element of the redo stack is popped and added to the undo stack.
* There is a read operation involved which when encountered prints all the elements of undo stack in reverse order.

7

FORMAT MODULE:

* Formatting data may be *qualitative* (e.g., [font family](https://www.computerhope.com/jargon/f/fontfami.htm)), or *quantitative* (e.g., [font size](https://www.computerhope.com/jargon/f/font-size.htm),or [color](https://www.computerhope.com/jargon/c/color.htm)). It may also indicate a style of *emphasis* (e.g., [boldface](https://www.computerhope.com/jargon/b/bold.htm), or [italics](https://www.computerhope.com/jargon/i/italic.htm)), or a style of *notation* (e.g., [strikethrough](https://www.computerhope.com/jargon/s/striketh.htm), or [superscript](https://www.computerhope.com/jargon/s/superscr.htm))
* Few options that can be included are: font-style,normal,italic ,bold,font size, bullets and numbering,etc..
* The java libraries that will be included

  for the same will be:

1) java.awt.event.\*

  2) javax.swing.\*

3) java.awt.\*and a few others.

* The JButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits AbstractButton class. We have used JButton for font colour.
* JComboBox is a part of Java Swing package. JComboBox inherits JComponent class . JComboBox shows a popup menu that shows a list and the user can select a option from that specified list . JComboBox can be editable or read- only depending on the choice of the programmer .We are using JComboBox for font box which is used  for choosing the size of text.
* We have created an array of strings to add all the fonts used in our text editor.

VIEW MODULE

* View module consists of Find/Replace , Spell Check, Zoom In, Zoom Out and Theme options
* For Spell Correction , we are using hash table, dictionary and JTextComponent from the Swing framework.
* For Find and Replace, we are creating a Label with options to Find, FindNext,Replace,Replace all and cancel.
* User can enter the text to be found and replaced in JTextField. After which, the index of entered text in Find will be looked and selected in JTextArea and then will be replace by method JTextArea.ReplaceSelection.

**8**

* For Zoom in and Zoom Out we’ll use JPanel and implement the MouseWheelListener in order to detect when the user rolls the mouse.
* For Find/Replace , we are using Swing,  java.awt.event.ActionListener and Action Event which will use methods of JTextArea to perform actions.
* For spell-checker, we're using our own dictionary.txt file .
* So, first we'll insert all the words from the dictionary file into a hash table. Then we'll start reading the text entered by user in the JTextArea. If the user entered word exists within the hash table, then that word is spelled correctly.
* Otherwise, it's again checked by removing all the punctuation characters.
* But if still the word is not found , then SpellingSuggest class is called which uses the EDIT DISTANCE  method to suggest the most probable word for the error.
* After the suggestion, we find the index of the wrong word and with the help of index we select the word in JTextArea and replace it by the suggested word.

SKETCHING DOODLES

Features Available:

* Draw strokes with pencil tool and set thickness.
* 11 Quick color selection buttons.
* Provided with Color Picker (Swatches, HSV, HSL, RGB, CMYK) using JColorChooser.
* Small size, easy-to-understand code.
* Save, Save As, Load buttons to Save/Load you work using Interactive JFileChooser.
* Undo, Redo
* Nimbus Look and Feel theme.
* Set Canvas size. ( Minimum recommended: 900x800)
* Enter Text

9

* For Sketching , we’ll use the package org.eclipse.swt.graphics contains classes that allows management of graphics resources.
* The subclass org.eclipse.swt.widgets.Canvas has a number of style bits that can be used to affect how painting occurs.
* Java GC has number of methods that allows to draw lines , shapes and text and fill colors. Java GC has number of methods that allows to draw lines , shapes and text and fill colors.
* We’ll be also using  presaved dataset to search for doodles by typing it’s category.

**10**

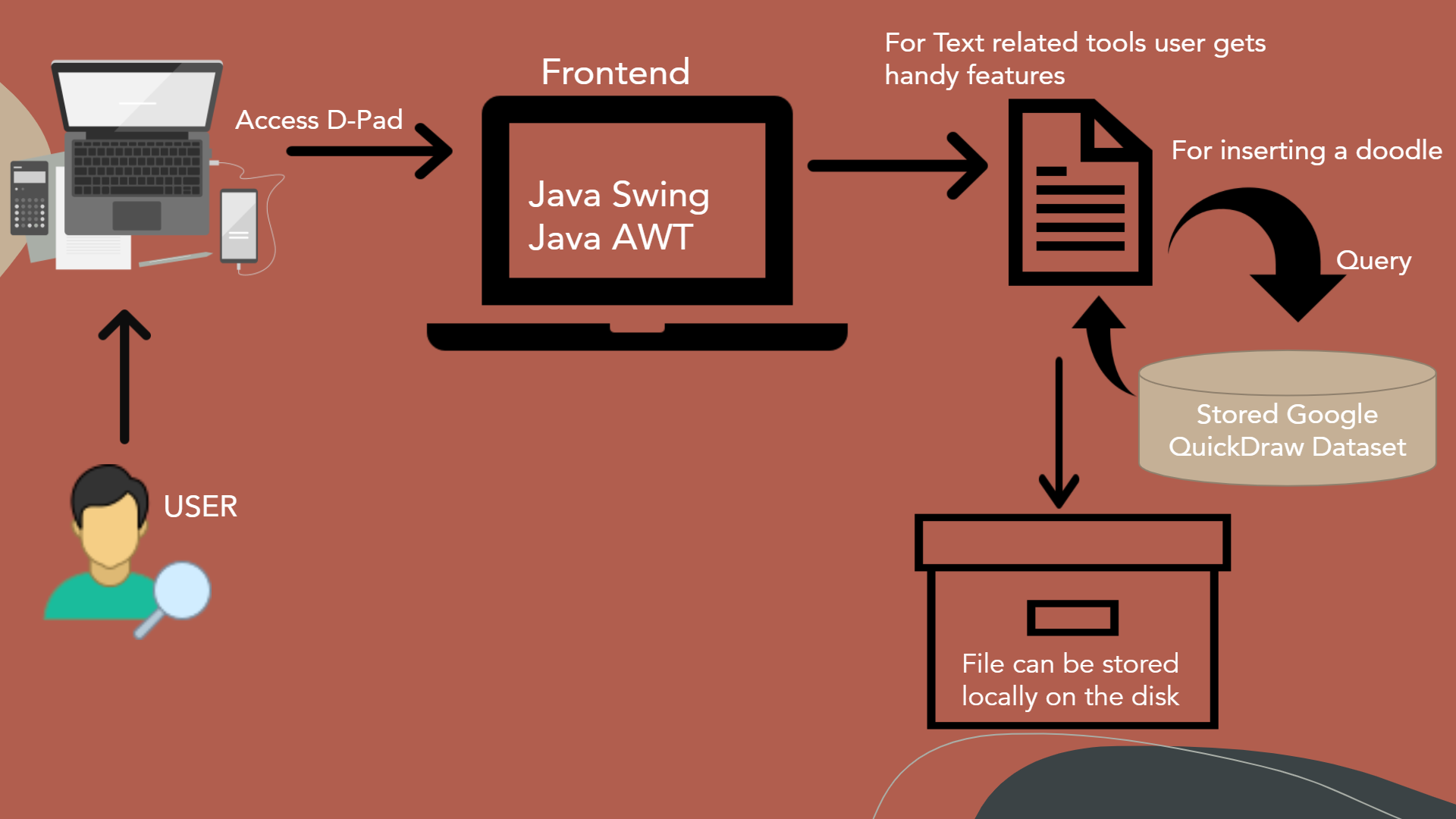
**SYSTEM DESIGN AND IMPLEMENTATION**

**INTRODUCTION**

In this section we are going to show our system architecture diagram which describes the modules that are described and tell about the hardware and software requirements of the project above and then we’ll elaborate our module implementation and discuss the module workflow diagram which shows the overall working of our text editor.

**11**

**SYSTEM ARCHITECTURE DIAGRAM**



*system architecture diagram - figure no 1.1*

12

**HARDWARE AND SOFTWARE REQUIREMENTS**

SOFTWARE:

For Text Editor:

* Java IDE
* Java Swing
* Java AWT
* Java UTIL

For Doodling:

* Java SWT(Standard Widget Toolkit)
* Google QuickDraw Dataset

HARDWARE:

* Intel Pentium III cpu or higher
* Windows 7 (32 or 64 bit) or above for full functionality
* Internet Connectivity

13

**MODULE WORKFLOW EXPLANATION**

Once the user opens the D-PAD Text Editor A typing area with text formatting operations will pop up. Either you can start writing text in the typing area and create a new .txt file or the user can open an existing .txt file form the option open given in the file menu section and edit the file further.

The JTextArea is used to create the complete typing area.

In this first we  specified the size of the text area as 500\*500 and used flow layout for the arrangement of the buttons and

used line wrap to avoid the usage of double scroll bars and make the text systematic.

FileChooser has been used in the open and save function in the file section.

JTextArea has been used to create the complete typing arera.

The JMenuBar and JMenuItems from the Java Swing library have been used to create the menu bar and the menu items which consists of various text editing option and a draw tool button.

The D-PAD text editor provides the user with various options to edit their texts,

Once the user enter some text in the given text area the user can:

1. Change the font of the text
2. Change the size of the text
3. Change the colour of the text
4. Change the background colour of the Text Area
5. By using the spell checker the user check for spelling errors
6. And the user can also use the Find and Replace feature to navigate a particular word and replace it.

This was all about editing of the text now if the user wants to create or add some doodles along with the text the user the can use the drawing tool button.

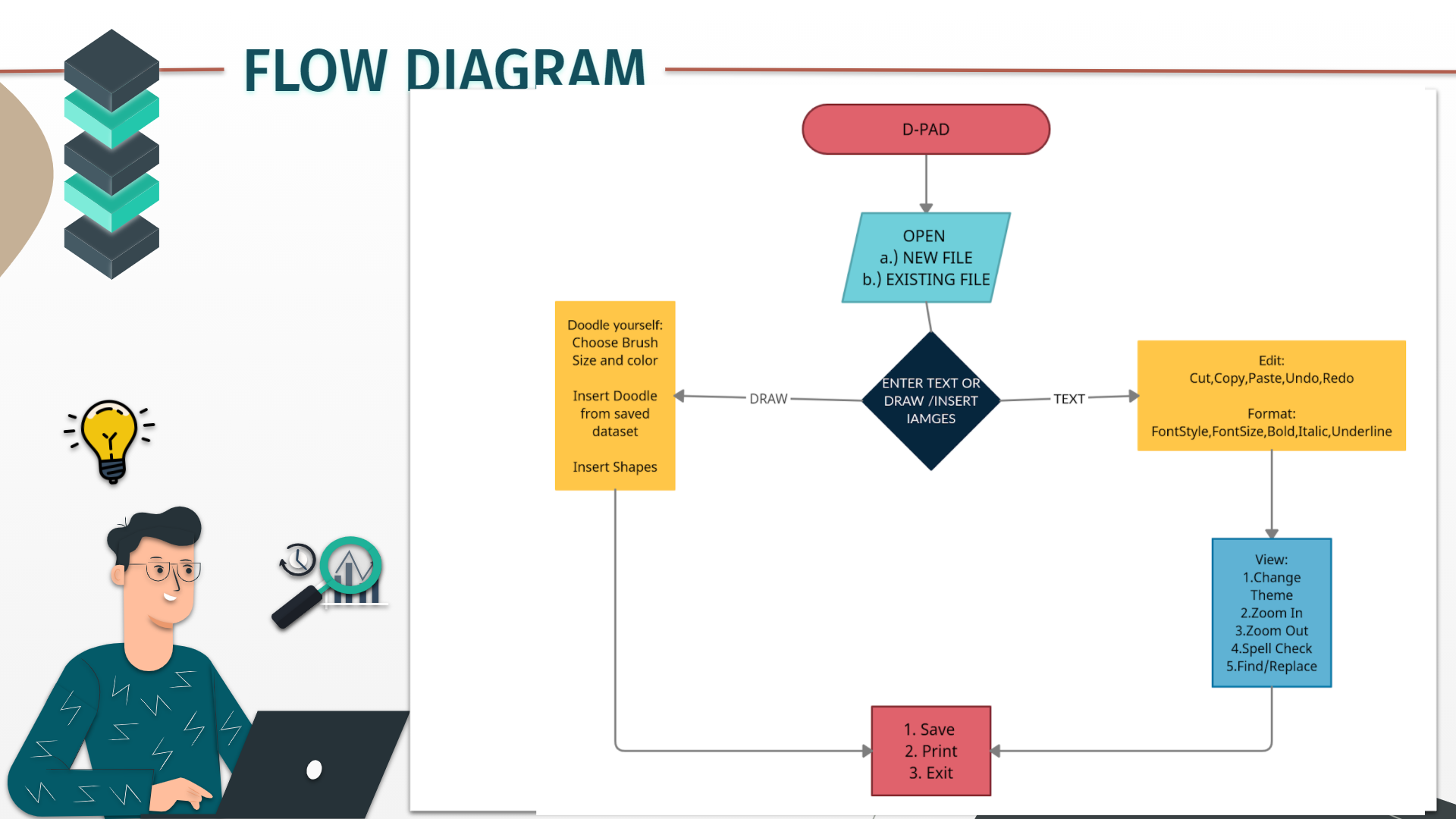
14

The drawing tool button takes the user to a canvas area where the user can either sketch doodles by using different brushes and colours or the user can search for various doodles.

To search for doodles first the user has to click on the doodle icon, this will display around 40-50 categories of doodles. The user can choose any of the categories this will display around 4-5 images of that category. From here the user can choose any of the image and add it to the canvas.

Once the user is done with all the text editing and adding dooldes. They can save the work using the save option in the file section this will pop save as dialog box from where the can choose the directory where they want to save the file name their document and click save.

This will save you work in the mention directory path.



*module workflow diagram - figure no 1.2*

**15**

**FUTURE ENHANCEMENTS AND CONCLUSION**

**NOVELTY OF THE PROJECT**

Dpad will have a featured doodle function enabling users to insert desired images in their documents furthermore providing users with pen like functionality to pour their imagination onto the pad.

On fruition, Dpad has a scope to transform the application into website enabling future expansion alongside extended improvisation and modification.

**16**

**REAL TIME USAGE**

Text Editor is not only related to programming but it is a software through which you can write or edit some text.

As far as programming is concerned you have to write or code your programme in a text editor and then you have to compile your code using a compiler.

1)It can be used to pen down thoughts.

2)We can take notes in class for a lot of subjects when teachers are teaching in pace and we have to keep up or take the summary / important points.

3)Using doodle will enhance our writing experience and will make the notes more attractive and understandable.

**17**

**FUTURE SCOPE OF THE PROJECT**

This project will largely be used by Everyone is attending online classes and meetings and there are times when we cannot write all the points or some symbols such as pi, epsilon, infinity,etc. are not even present on the keyboards.Therefore in such scenarios our project could be a huge breakthrough. We are adding more and more symbols in our dataset for the user’s convenience.

**RESULT AND** **CONCLUSION**

This project will largely be used by students because now-a-days,everyone is attending online classes and meetings and there are times when we cannot write all the points or some symbols such as pi, epsilon, infinity,etc. are not even present on the keyboards.Therefore in such scenarios our project could be a huge breakthrough. We are adding more and more symbols in our dataset for the user’s convenience.

18

**REFERENCES**

1. <https://www.geeksforgeeks.org/implement-undo-and-redo-features-of-a-text-> editor/
2. <https://www.programmingnotes.org/3144/c-simple-spell-checker-using-a-hash->table
3. https://www.geeksforgeeks.org/java-swing-jtextarea/
4. https://www.geeksforgeeks.org/gap-buffer-data-structure/

19