KOLHAPUR INSTITUTE
OF TECHNOLOGY'S
COLLEGE OF
ENGINEERING
KOLHAPUR

A
Mini Project Report On

“KEYBOARDING”

Submitted To

Department of Computer Science & Engineering, KITCoEK, Kolhapur

In Fulfillment of the syllabus of S.E.C.S.E examinations

Submitted By

Roll No.	Name
98	Sourabh D. Khanaj
100	Shital B. Kallole
101	Aditya R. Karmalkar

Under The Guidance Of
Prof. S. S. Kale
(M.E. CSE)

Department Of Computer Science & Engineering
KIT’S College Of Engineering, Kolhapur.

Year
2016-17

CERTIFICATE

KIT'S COLLEGE OF ENGINEERING, KOLHAPUR

This is to certify that

Name	Roll No.
Sourabh Dattatray Khanaj	**98**
Shital Basavraj Kallole	**100**
Aditya Raju Karmalkar	**101**

have successfully completed the mini project on project entitled "KEYBOARDING", in the partial fulfillment of the syllabus of examinations **o**f S.E. C.S.E Part-II of KIT's College of Engineering, Kolhapur in the of the academic year 2016-17.

Project Guide External Examiner H.O.D(CSE)

Kolhapur Institute of Technology's
College of Engineering, Kolhapur.
Year
2016-17

ACKNOWLEDGEMENT

We would like to express our deep gratitude to Prof. S.S.Kale & Head of Department (CSE), **Dr. P. S. Patil** for their constant encouragement and belief in us. Their guidance and attention throughout the project work has been of immense help to us.

We express our sincere thanks to all the teaching and nonteaching staff and all those who have directly or indirectly helped in making project a success.

Sincerely by

Name	Roll No
Sourabh D. Khanaj	98
Shital B. Kallole	100
Aditya R. Karmalkar	101

ABSTRACT

This project is about counting the minimum number of keystrokes required to type a given message. Suppose, if we want to type any message then the number of keystrokes required are equal to number of characters present into the message. But, there is a hardware which has only five hardware buttons and other characters are present virtually i.e. on the screen, in this situation we can use this project.

CONTENTS

1. INTRODUCTION

1.1 Motivation

In the 21st century, we are living in the computerized world and our lives become faster even than the airflows. Everyone is rushing within bit of minutes of time, in each routine activity. **"Time is precious to SAVE"** and keeping this mind, almost all the manually working system are computerized.

All this keeping in mind we build an idea to create the "Keyboarding" which helps the user to tell the minimum keystrokes required for type the message.

1.2 Problem Definition

To find the path which contains minimum number of keystrokes to type a text for virtual keyboard.

1.3 Objective of Project, Need, Scope

The very main and the basic objective of our Keyboarding System are to find the minimum keystrokes required to type a simple message. And it can be accessed in easily and efficient manner. It is very essential and important nowadays to computerize the system on time before it's too late.

1.4 Advantages of project

1) Keyboarding is used for small pocket size electronic devices.
2) It is user friendly program.
3) Finds absolutely minimum keystrokes to type any message.

1.5 Limitations of Project

1) Valid Character Allowed in Virtual Keyboard i.e. A-Z , 0-9, -,*
2) Text to be typed must not empty and don't contain * because asterisk(*) representing ENTER Key.

2. LITERATURE SURVEY

2.1 Introduction

If we want to count the number of keystrokes required to type a text message then that time we may think that this is equal to number of characters present into the message, but this is correct only if one keystroke generates one character. But the electronic device or with pocket size devices the possibilities for typing text are often limited because with less number of hardware keys may require multiple keystrokes to type characters because the characters are displayed virtually on the screen. Some devices provide only a few buttons significantly fewer than the number of letters in alphabet for such devices several strokes may be needed to type a single character.

One mechanism to deal with this limitation is a virtual keyboard displayed on the screen with a cursor that can be moved from key-to-key to select characters. Four arrow button control the movement of the cursor and when the cursor is positioned over an appropriate key pressing the fifth button selects the corresponding character and appends it to the end of the text. At last we terminate the text by pressing the key.

2.2 Existing System

The existing system for this project is just find out only minimum keystrokes ad display to the user. It is difficult for users who does not know about the system working and its relative techniques.

2.3 Disadvantages of Existing System

1) At the time of creation of keys, user can create one key only once.

2) Difficult to understand to the users who are non-familiar with the computer system.

2.4 Proposed System

Proposed system display the minimum keystrokes and also displays the path. Because of that the users can understand the system very well. Also we can produce the user manual contains the detailed information about our system.

3. REQUIREMNT ANALYSIS

3.1 Software Requirement Specification

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete **“KEYBOARDING”** by defining the problem statement in detail.

3.1.1 User Requirement

1) Users have to create own keyboard.

3.1.2 Software Requirement

Software Interface:

1) Any version of Linux Operating System
2) VMware Workstation

3.1.3 Hardware Requirement

1) Processor – Pentium 2 and above

2) RAM – Minimum 512MB

3) Hard Disk Space – Minimum 10 MB

4. DESIGN

4.1 Flowchart

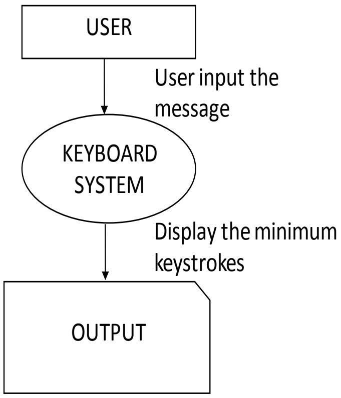


Fig. DFD(Referenced from Pankaj Jalote)

4.2 Structure Used

```
struct linked
{
        char link[4];
};



struct symbolTable
{
        Short int status; // character is present or not (1=present, 0=absent)
        Short int startRow; //starting row index
        Short int startCol;  // starting column index
        Short int endRow; // ending row index
        Short int endCol;  //ending column index
        Short int type; //type of the character
        Short int length; //number of neighbor character
        struct linked connect; //neighbor characters are stored.
    }symbol[38];
    //Symbol Table id used to maintain the information about all the characters
    present in keyboard
    //38 is size because the allowed character = A-Z(26) , 0-9 (10) , - (1), *(1)
```

4.3 Modules

- Main()

  Calling of following functions:

  A. input()
  B. tableAnalysis()
  C. tableSymbolType()
  D. tableSymbolLength()
  E. validation()
  F. process()

- input()

  -calling of instruction()

  -We take the keyboard structure from user and then keyboard contents and then message to be typed.

- tableAnalysis()

  - We find out starting index and ending index of each character present in keyboard.

- tableSymbolType()

  - Calling of findPosition().
  - In this function we find out the type of each key. Type means it is made of single square (TYPE=1) or it is made of multiple squares.
  - If it is made up of multiple squares then

    1) Horizontal key

       If there is different character at left side then TYPE=2.

       If there is different characters in left and right both side then TYPE=3.

       If there is different character at right side then TYPE=4.

    2) Vertical Key

       If there is different character at lower side then TYPE=5.

       If there is different characters in lower and upper both side then TYPE=6.

       If there is different character at upper side then TYPE=7.

- tableSymbolLength()

  We find the all neighbors of each characters present in keyboard and link all that neighbors to that character.

- validation()

  -In this function we are going to check that all character present in keyboard are must be allowed characters i.e. A-Z , 0-9, - ,* .

  - We also check that asterisk(*) character must be prestnt in keyboard because it is used to indicate end of string.

  - We check the character present in message is not containg asterisk(*) and that character present in keyboard.

- process()

  Calling of following functions:

  - basicInitializationInitial()
  - basicInitializationString()
  - findPosition()
  - findActualString()
  - recursion()

5. IMPLEMENTATION & RESULTS

5.1 Explation of sub functions

instruction()

It is used to display the instructions.

findPosition()

It takes two integer arguments i & j where i indicate row number and j indicate column number.

It gives the position from symbol table of character present at index i & j in keyboard.

basicInitializationInitial()

To get the starting and ending index of the character which is pointed by cursor currently.

basicInitializationString()

To get the starting and ending index of character which is present in message and which is currently pointed stringpoint(it is initially pointing to first character present in message and as we process it is implementing).

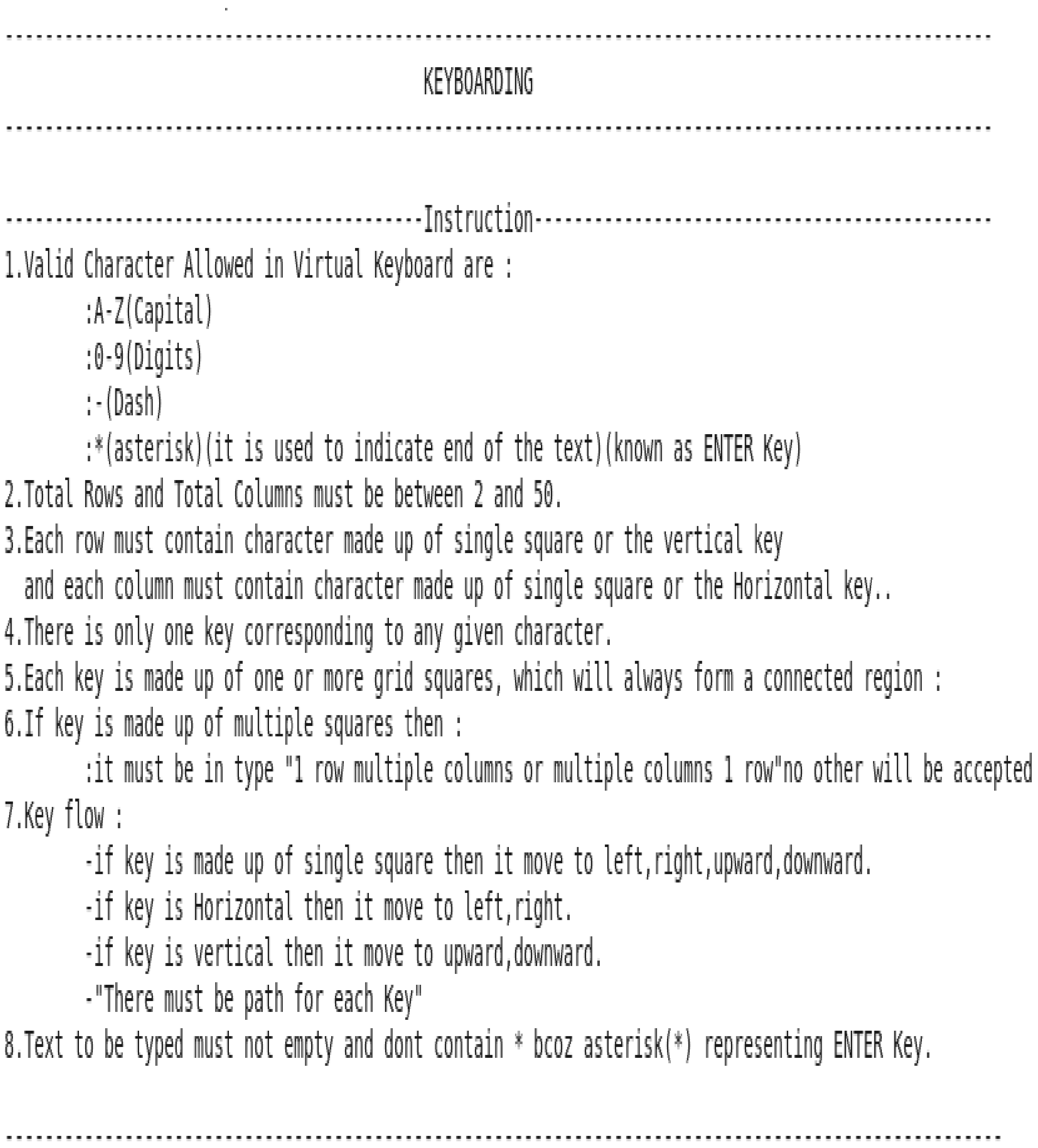
findActualString()

It is used to get the character pointed by position from symbol(symbol table).

recursion()

This method is used to find the path which contains minimum keystrokes to type the message. In this method recursion is done.

5.2 Output Screens



```
-----------------------------------------------------------------------------------------------
                                        KEYBOARDING
-----------------------------------------------------------------------------------------------


---------------------------------------Instruction---------------------------------------------
1.Valid Character Allowed in Virtual Keyboard are :
        :A-Z(Capital)
        :0-9(Digits)
        :-(Dash)
        :*(asterisk)(it is used to indicate end of the text)(known as ENTER Key)
2.Total Rows and Total Columns must be between 2 and 50.
3.Each row must contain character made up of single square or the vertical key
  and each column must contain character made up of single square or the Horizontal key..
4.There is only one key corresponding to any given character.
5.Each key is made up of one or more grid squares, which will always form a connected region :
6.If key is made up of multiple squares then :
        :it must be in type "1 row multiple columns or multiple columns 1 row"no other will be accepted
7.Key flow :
        -if key is made up of single square then it move to left,right,upward,downward.
        -if key is Horizontal then it move to left,right.
        -if key is vertical then it move to upward,downward.
        -"There must be path for each Key"
8.Text to be typed must not empty and dont contain * bcoz asterisk(*) representing ENTER Key.


-----------------------------------------------------------------------------------------------
```

```
Enter Total Rows and Columns In Keyboard : 5 5

Enter Keyboard as per the given structure :
A B C D D
E F H R I
S 0 4 T L
M N Z Z O
7 * * * -
Enter string to be typed:CRISTIANO07
--------------------------------------------------------------------------------
        Character       Strokes         Path Which Contain Minimum Strokes
--------------------------------------------------------------------------------
            C              3             A-> B-> C
            R              3             C-> H-> R
            I              2             R-> I
            S              6             I-> R-> H-> F-> E-> S
            T              4             S-> 0-> 4-> T
            I              3             T-> L-> I
            A              5             I-> D-> C-> B-> A
            N              5             A-> B-> F-> 0-> N
            O              3             N-> Z-> O
            0              4             O-> Z-> N-> 0
            7              4             0-> S-> M-> 7
            *              2             7-> *
--------------------------------------------------------------------------------

Total Minimum Strokes Required to type String "CRISTIANO07" using Virtual Keyboard is :
 44
```

6. TESTING & VALIDATION

6.1 Introduction

Software Testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. Quality is not an absolute; it is value to some person An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

6.2 Design of test cases and scenarios

Sr. NO	USER ACTION	EXPECTED RESULT	OBSERVED RESULT	TEST RESULT
1	User is enter rather than A-Z, 0-9, - ,*	Should display an error message “Character in keyboard must be A-Z or 0-9 or – or *”	Same	Pass
2	User enters 0 rows and 0 columns	Should display an error message “Total rows and Total column must be between 2 and 50”	Same	Pass
3	If User enter the keys without * key.	Should display an error message “Asterisk (*) must	Same	pass

		present in keyboard because it represent ENTER key.”		
4	When user enter message and character in string are not present in keyboard.	Should display an error message “character present in message but not present in keyboard”	Same	pass
5	When user enter message and characters in message are not allowed	Should display an error message “character is not allowed in message”	Same	Pass
6	When user enter non-unique keys in one row of keyboard	Should display an error message “error in row”	Same	Pass

7. CONCLUSION

This project is useful for small pocket size electronic devices which has less number of hardware keys (LEFT, RIGHT, DOWN, UP arrow and SELECT key).

We can count the number of required minimum keystrokes and display the path for minimum keystroke using.

REFERENCES

- ACM-ICPC 2015 Problems Paper