STUDENT SERVANT PROJECT

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

D. Prajwal Reddy– 1602-19-737-086

Sriram – 1602-19-737-114



**Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31 202**

**Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University)**

**2020**

**Hyderabad-500 031 Department of Information Technology**

**DECLARATION BY THE CANDIDATES**

We, PRAJWAL REDDY and SRI RAM bearing hall ticket numbers, 1602-19-737-086 and 1602-19-737-114 respectively, hereby declare that the project report entitled “STUDENT SERVANT” is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

D. Prajwal Reddy – 1602-19-737-086

Sriram – 1602-19-737-114

(Faculty In-Charge) (Head, Dept. of IT)

**AKNOWLEDGEMENTS**

Our Mini Project, Student Servant, would not have been any successful or meaningful without the help of several people. We are extremely thankful to our college, Vasavi College of Engineering, for providing us with the opportunity to ideate and implement our project, “STUDENT SERVANT”.

We would like to express our gratitude to Ms. Prasanna Ma’am, Assistant Professor, Department of Information Technology, Vasavi College of Engineering and Ms. Divya Ma’am, Assistant Professor, Department of Information Technology, Vasavi College of Engineering, for their acceptance, esteemed guidance, moral support and invaluable advice provided by them for the success of the Mini Project right from the beginning.

We are grateful for the time and interest you have shown for us to move with the project in the right direction.

Sincerely,

**D. Prajwal Reddy 1602-19-737-086**

**L. Sri ram 1602-19-737-114**

**ABSTRACT**

In this day and age, from STUDIES to sales, everything is using online platforms. So, the time a student has to spend in front of a computer is increasing.

\*OUR MAIN OBJECTIVE IS TO BUILD AN APPLICATION FOR MANAGING NOTES WHILE ONLINE CLASSES ARE ONGOING AND EVEN AFTER.

\*THROUGH THIS APPLICATION, ONE HAS QUICK ACCESS TO write, edit and read the notes everyday FOR VARIOUS SUBJECTS.

**TABLE OF CONTENTS**

1. Introduction………………....................................................7
2. Technology…………………………………………………9

a. Software Requirements

b. Hardware Requirements

1. Proposed Work……………………………………………10

3.1Design…………………………………..……………..10

3.2 Implementation………………………..…………..….11

3.3Testing……………………………………….………..27

3.3.1 Test Plan………………………………….27

3.3.2 Use Cases………………………….…...…28

1. Results……………………………………………………30
2. Additional Knowledge Acquired…………………………40
3. Conclusion of Discussion and Future Work………..…….41
4. References……………………………………………..….42

# INTRODUCTION

## ABOUT THE PROJECT

“STUDENT SERVANT” is a console-based C Project which keeps the notes of classes of students of the institution. The program is run by the student who can add, save, edit, delete, and find notes and view or study according to the need.

## 1.1 PROJECT DOMAIN

The Note Writing is a very big part of a Student's life. Right from initial stages of schooling to completing graduation, It is necessary to attend the classes and prepare one's own notes despite the presence of books regarding that subject and topic. Notes were earlier written on note-books (papers) manually. This system is still in practice. But now-a-days, we are noticing some of the disadvantages of this. Since Notes are heavy, it is hard to carry them from one place to another, especially to foreign. Books can be easily damaged, torn and pages can go missing. Sometimes, our own handwriting is ambigue or scribbled. Note-books are shared will often be found missing or unauthorized changes may lead to problems while referring to them or studying.

**Purpose**

Now-a-days, since everything is a digital data. Why not our knowledge or interpretation be automated to be saved and secured for all the future purposes.

The Main Aim of this project is to provide a servant to each student which does the job in a very accurate, understandable and appreciable way. This Project is personified and not complicated at all.

**Benefits**

* To make it easy to write or read notes and store them safe.
* Quick access to find out notes written for which date.
* To provide easy and faster access of notes as per subject, topic, lecturer etc.
* To provide a student friendly environment of writing or editing notes.

**1.2 FEATURES**

In this project the student, who is the main primary user, can add, save, modify/edit, search, view and delete the notes of different subjects and topics.

**1.2.1 Add Notes**

The student can write notes through add menu per the specified subject, the date the class taught, time period of it and the lecturer along with his/her notes.

**1.2.2 View Notes**

The student can search the notes written for a certain date and read from them.

**1.2.3 Edit Notes**

The student can modify/edit the detatils of any type of the notes pre-written by searching and will be saved to the requirements of the student with changes applied.

**1.2.4 Delete Notes**

The student can delete the notes of the student written on any date according to requirements.

# TECHNOLOGY

Every Programs needs some sort of pre-installed or invented software in the computer. All computer software needs certain hardware components or other software resources to be present, in order for computers to be used efficiently. These prerequisites are known as System Requirements. Within this, we have two types – Software Requirements and Hardware Requirements.

## a. SOFTWARE REQUIREMENTS

Software Requirements deal with defining the software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application or the Mini Project we ideated and coded. These preconditions are generally not included in the software installation package and need to be installed separately.

In order to use CODIAC, one should have the following:

* **Operating System:** Windows 7 or above
* **C Compiler:** GNU Compiler Collection (GCC)
* **Editor:** Any text editor ( Note Pad/ Note Pad++..) .

## b. HARDWARE REQUIREMENTS

Hardware requirements refer to the common set requirements defined by any operating system or software application and are usually the physical computer resources. In this, we look into the architecture, processing power, memory, secondary memory, display adapter and peripherals.

In order to use this project, one should have the following:

* + - **Processor:** Intel Pentium processor and above ( or equivalent..)
    - **Memory:** 2 GB RAM and above

1. **PROPOSED WORK**

**3.1 DESIGN**

The Design of this Mini Project has been done in a way to first look at the process of the flow or functioning and add the modules with their basic properties to the project.

**3.1.1 USE CASES**

**Diagram

Description automatically generated**

**3.1.2. FLOWCHART:**

**Diagram

Description automatically generated**

## 3.2 IMPLEMENTATION

## DESCRIPTION OF MAIN MODULES

**3.2.1 LOGIN**

The student has to login with the institution issued username and password. If the password or the username are attempted incorrectly for thrice then it shows a message of wrong password or user name and exits. If the credentials were correct then the student will be given access to view at the actual menu of the student servant project.

MAIN MENU DISPLAYS THE FOLLOWING OPTIONS AND ASKS TO CHOOSE

**3.2.2 ADD NOTES**

The student has to choose this option and it takes to the adding notes sub menu. In this one has to enter the date for which he/she wants to write the notes for. Then He/She has to enter the period time, subject, topic, lecturer name and notes. After finishing, one can tell whether he wants to continue writing more notes or not and based on that same process continuos or moves to main menu.

**3.2.3 VIEW NOTES**

The student can view the notes he/she has already written through the viewing sub menu. One has to enter the date of the notes he/she wants to read and then the notes of that date for various subjects, topics will be displayed. One can continue to view the notes of other dates or can go back to main menu.

**3.2.4 EDIT NOTES**

The student can edit the already written notes through the editing sub menu. One has to enter the date of which he wants to edit notes and then select whether he/she wants to edit time period or topic or notes etc and after editing can choose if he wants to edit more notes and the no of notes edited will be displayed. One can read the changes in the view menu per the same date.

**3.2.5 DELETE NOTES**

The student can delete the notes he/she wishes are no longer necessary and does not want to spend memory for them from the deleting menu. One has to enter the date of the notes he wants to enter and all the notes written on that date will be deleted. One can continue doing so or can go back to main menu.

**3.2.6 EXIT**

The student from the main menu upon choosing the exit option, the program gets terminated.

**CODE:**

From the use cases, we have implemented this project by dividing the work into the following modules – Start, Login, Add Notes, View Notes, Edit Notes, Delete Notes and Exit is to terminate the application.

**START**

MAIN:

Graphical user interface, text, application

Description automatically generatedText

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

LOGIN

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

ADD NOTES

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

VIEW NOTES

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

EDIT NOTES:

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

DELETE NOTES:

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**ALGORITHM/LOGIC:**

CODE FOR VIEWING NOTES:

do

{

printf("\n\tENTER THE DATE OF THE NOTE TO BE VIEWED:[dd-mm-yyyy]:");

fflush(stdin);

gets(filename);

fpte = fopen ( "filename", "rb+" ) ;

if ( fpte == NULL )

{

puts ( "\nTHE RECORD DOES NOT EXIST...\n" ) ;

printf("PRESS ANY KEY TO EXIT...");

getch();

return ;

}

system("cls");

printf("\nTHE NOTE FOR %s :",filename);

while ( fread ( &e, sizeof ( e ), 1, fpte ) == 1 )

{

printf("\n");

printf("\n PERIOD TIME: %s",e.periodtime);

printf("\n SUBJECT: %s",e.subject);

printf("\n TOPIC: %s",e.topic);

printf("\n LECTURER: %s",e.lecturer);

printf("\nNOTE: %s",e.note);

printf("\n");

}

printf("\n\nWOULD YOU LIKE TO CONTINUE VIEWING FOR OTHER DATES...(Y/N):");

fflush(stdin);

scanf("%c",&choice);

}

while(choice=='Y'||choice=='y');

fclose ( fpte ) ;

return ;

**ALGORITHM:**

1. Reads the date of the note to be viewed and opens its corresponding file in appropriate mode.

2. If there are no pre written notes for the date, then displays file not found.

3. If there is notes for the date, the uses a file pointer to read the file in the structured order.

4. Then Prints the period time, topic, lecturer, subject, notes etc

5. Asks whether to continue viewing more notes, if character entered is y or Y, then similar process continues.

6. Else Closes the File pointer and returns or exits.

**GITHUB LINK AND FOLDER STRUCTURE:**

LINK:

<https://github.com/Prajwalreddydornala/mini_project1>

FOLDER STRUCTURE:

Graphical user interface, application

Description automatically generated

## 

## 3.3TESTING

Testing is a method to check whether the actual product matches the expected requirements and to ensure that the product is defect-free. This process involves execution of various parts of the product either using manual or automated tools. The purpose is to identify errors, gaps or missing requirements in contrast to the actual requirements.

**3.3.1 TEST PLAN**

It is important to know what are the possible test cases. We approached testing our console application by analysing each module separately ( add notes etc). First, we coded the requirements and then manually tested each feature present in the module to cover any gaps that might occur as the better implementation without confusions should be not compromised.

**3.3.2 USER TEST CASES**

The user STUDENT has to go through different modules like Login, Add Notes, View Notes, Edit Notes, Delete Notes. Below are the testcases which we have compiled together manually.

* + - 1. **3.3.2.1 Login**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID:** TC01 | | **User Case ID:**  UC01 |
| **Test Case Title:**  Login | |
| **Test Case Description:** Student attempts to login into the application with incorrect credentials for maximum of three times | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays prompt for user to enter username  And password(predefined by institution) | System should display “Invalid credentials”. | The details were attempted incorrectly for three times the application terminates. |

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID:** TC02 | | **User Case ID:**  UC02 |
| **Test Case Title:** Login | |
| **Test Case Description:** Student attempts to login into the application with the correct credentials | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays prompt for user to enter username and password (within 3 tries) | Enter into the application. | Enters into the main menu of application. |

* + - 1. **3.3.2.2 ADD NOTES**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC03** | | **User Case ID:**  UC03 |
| **Test Case Title: Add Notes** | |
| **Test Case Description:** Student add the details of student/employee | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the details of date and others. | System saves the notes successfully. | Saves successfully. |

* + - 1. **3.3.2.2 View Notes**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC04** | | **User Case ID:**  UC04 |
| **Test Case Title: View Notes** | |
| **Test Case Description:** Student search the details of notes with details of dates that does not exist | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the details to search the notes. | System displays to Records not found. | System displays Records not found. |

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC05** | | **User Case ID:**  UC05 |
| **Test Case Title: View Notes** | |
| **Test Case Description:** Student enters the date for which notes exist and views. | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the details to search the notes. | System displays to Notes. | System displays the details. |

* + - 1. **3.3.2.3 EDIT NOTES**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC06** | | **User Case ID:**  UC06 |
| **Test Case Title: Edit Notes** | |
| **Test Case Description:** Student tries to edit the notes of dates that does not exist. | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System does not display to enter the details to edit the notes. | System displays to file not found. | System displays Notes not found. |

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC07** | | **User Case ID:**  UC07 |
| **Test Case Title: Edit Notes** | |
| **Test Case Description: Student search the notes with dates if file exist.** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the details to search and view the notes. | System displays edited successfully , saves changes in notes successfully. | System saves Note edited Successfully. |

* + - 1. **3.3.2.4 DELETE NOTES**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC08** | | **User Case ID:**  UC08 |
| **Test Case Title: Delete Notes** | |
| **Test Case Description:** Student tries to delete notes that are not even added. i.e., that file does not exist | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the details to delete the records. | System displays to file not found. | System displays Notes not found. |

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC09** | | **User Case ID:**  UC09 |
| **Test Case Title: Delete Notes** | |
| **Test Case Description:** Student enters to delete the notes with date details which if note exist. | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the details to delete the records. | System displays Note deleted successfully. | System displays Notes deleted Successfully. |

# 4.RESULTS

# Text Description automatically generatedText Description automatically generatedGraphical user interface, text, application Description automatically generatedGraphical user interface, text, application, Word Description automatically generated

# Graphical user interface, application, Word Description automatically generatedA picture containing text Description automatically generatedA picture containing shape Description automatically generatedText Description automatically generatedA picture containing text Description automatically generated

# Text Description automatically generatedText Description automatically generatedA picture containing shape Description automatically generatedText Description automatically generatedA picture containing shape Description automatically generated

# A picture containing text Description automatically generatedA picture containing text Description automatically generatedA picture containing shape Description automatically generatedText Description automatically generated with medium confidence

# 5.ADDITIONAL KNOWLEDGE ACQUIRED

This Project has been based on the real issues of us as students. In the process of coding different modules for this project, we have learnt the real life application of stacks since the viewing of notes is in such order. We have been introduced to libraries such as time.h and windows.h which will be helpful to know the time and apply it on main menu. We have expanded our boundaries to know the importance of glorification by graphics which we have learnt during the process. We have been made as masters in file handling and working with any complicated case where files has played or will play an important role. Also the real life necessary of coding in C based on our ideas has been a wonderful Journey for us.

We , have leant to co-ordinate and share work as a team of two and successfully did complete justice to our parts through mutual understanding and knowledge and confidence over the project.

# 6.CONCLUSION AND FUTURE WORK

To conclude, this application is prominently useful and can be handled easily by a 10 year old as well. This application introduces digital storing for a basic child and also gives immense support of portability of self written notes even for a 30 year old.

The Future work on this project is to lead to create an android application as well as an HTML web page. This Project can be further developed to give access to our teachers to look into our notes and correct them by sharing the HTML link.

**7. REFERENCES**

1. C Language Documentation: [https://docs.microsoft.com/en-us/cpp/c- language/?view=msvc-160](https://docs.microsoft.com/en-us/cpp/c-language/?view=msvc-160)
2. Stack Overflow (for debugging errors): <https://stackoverflow.com/>

THANK YOU.