**CALTALK!**



**(Voice Command Based Calculator)**

***A Project Report submitted in partial fulfillment of the requirements for the award of the degree of***

# Bachelor of Computer Application

## By

|  |  |
| --- | --- |
| **Ms. Payal Goyal** | **(174200101)** |
| **Mr. Pradeep Raghav** | **(174200103)** |
| **Mr. Pradum Kumar** | **(174200104)** |
| **Ms. Pragati Dubey** | **(174200105)** |
| **Mr. Prashant Agnihotri** | **(174200108)** |
|  |  |

Under the Guidance of

**Mr. Mohd. Aamir Khan (PhD Scholar)** Department of Computer Engineering & Applications

**Institute of Engineering & Technology**

**GLA University Mathura- 281406, INDIA**

**April, 2019**

**Department of Computer Engineering and Applications GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road,**

**P.O. – Chaumuhan, Mathura – 281406**

**U.P (India)**

**Declaration**

We hereby declare that the work which is being presented in the B.C.A. Project **“CALTALK (Voice Based Command Calculator)”**, in partial fulfillment of the requirements for the award of the ***Bachelor of Computer Applications*** and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of **Mr. Mohd Aamir Khan, PhD Scholar, Department of Computer Engineering and Applications.**

The contents of this project report, in full or in parts, have not been submitted to any other institute or university for the award of any degree.

|  |  |  |
| --- | --- | --- |
| Payal Goyal  (174200101) |  | Pradeep Raghav  (174200103) |
| Pradum Kumar  (174200104) |  | Pragati Dubey  (174200105) |
| Prashant Agnihotri  (174200108) |  |
|  | **Certificate** |  |

This is to certify that the above statements made by the candidates are correct to the best of my/our knowledge and belief.

#### Project Supervisor Date: 23 April 2019

(**Mr. Mohd Aamir Khan**)

#### Project Incharge Program Coordinator Head of Department

**(Mr. D.P. Yadav) (Mr. Narendra Mohan) (Prof. Anand Singh Jalal)**

**ACKNOWLEDGEMENT**

The satisfaction which accompanies the successful completion of the project is incomplete without the mention of a few names. We take this opportunity to acknowledge the efforts of the many individuals who helped us to make this project possible.

Firstly we would like to express our heartfelt appreciation and gratitude to our project guide **Mr. Mohd Aamir Khan, PhD Scholar, Computer Engineering & Applications Department.** His vision and execution aimed at creating a structure, definition and realism around the project and fostered the ideal environment for us to learn and do. This project is a result of his teaching, encouragement and inputs in the numerous meetings he had with us, despite his busy schedule.

We would also like to extend our immense gratitude to our Program Coordinator Mr. N………. respected Head of Department **Prof. Anand Singh Jalal** who allowed us to choose the topic for our dissertation.

The experience was novel one and we would like to thank all the people, who have lent their valuable time for the completion of the report. Without their consideration it would have been difficult to complete the report.

|  |  |
| --- | --- |
| Ms. Payal Goyal | (174200101) |
| Mr. Pradeep raghav | (174200103) |
| Mr. Pradum Kumar | (174200104) |
| Ms. Pragati Dubey | (174200105) |
| Mr. Prashant Agnihotri | (174200108) |
|  |  |

**ABSTRACT**

Caltalk performs all the tasks like scientific calculations, temperature conversions, unit conversions etc. Basically, we have different applications for different purposes like for scientific calculations, we have scientific calculators.

This is a desktop application which takes input by voice as well as by keyboard. It is helpful for those who can’t give input through keyboard or we can say helpful for physically disabled person. If the user gives command by voice then it simply recognize the voice and process it to produce the output but if the user gives command by input then first it convert text into voice then process it to produce the effective output.

It provides output in the form of text as well as in the form of voice.

**CONTENTS**

|  |  |
| --- | --- |
| **Topic**  Certificate Acknowledgement  Abstract | **Pg. no.** ii  iii  iv |
| List of Figures | viii |
| 1 Introduction | 1 |
| 1.1 Overview & Motivation | 1 |
| 1.2 Objectives | 2 |
| 1.3 Organization of Project Report | 2 |
| 2 Software Requirement Analysis | 3 |
| 2.1 Requirement Analysis | 3 |
| 2.1.1 Hardware Requirement | 4 |
| 2.1.2 Tools and Technologies | 4 |
|  | 4 |
| 2.2 Feasibility Study | 5 |
| 2.2.1 Technical Feasibility | 5 |
| 2.2.2 Operational Feasibility | 6 |
| 2.2.3 Economical Feasibility | 7 |
| 2.3 Analysis | 7 |
| 2.4 Summary of Modules | 8 |
| 3 Software Design | 9 |
| 3.1 DFD | 10 |
| DFD Level 0 | 11 |
| DFD Level 1.1 | 11 |
|  |  |
|  |  |
|  |  |
| 4 Implementation & User Interface | 14-15 |

|  |  |
| --- | --- |
| 5 Software Testing | 16 |
| 5.1 Testing | 16 |
| 5.2 Objectives of Software Testing | 16 |
| 5.3 Principles of Software Testing | 17 |
| 5.3.1 White Box Testing | 17 |
| 5.3.2 Black Box Testing | 17 |
| 5.4 Testing Fundamentals | 18 |
| 5.5 Testing Information | 18 |
| Conclusion | 19 |
| References | 20 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Figure Name** | | | **Fig. No.** | **Page** |
| **1** | Data Flow Diagram Symbols | | | Fig 3.1 | 10 |
| **2**  **3** | 0 level DFD  1 level DFD | | | Fig 3.2  Fig 3.3 | 11  11 |
|  |  |  |
|  |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |
|  |  | | |  |  |

# INTRODUCTION

**Present Problem Statement:**

There are many desktop applications which performs Single task at a time but these are found unreliable when we want to perform many operations, then we have to download multiple applications in our desktop.

We have combined many applications such as Number conversions, scientific calculations, financial calculations etc. in a single application.

In existing applications, we can’t give input by voice, but in CalTalk we can give input by voice as well as output in the form of audio.

**Proposed System:**

In CalTAlk, we can perform mathematical, scientific and other tasks like financial calculations, unit conversions and temperature conversions in a single application. In this, we can give input by voice or text both and output can also receive in the form of voice or text both.

In this, there is no need to write anything after given the command. When we give the input, it directly produces the output.

This application is implemented on the desktop with no additional hardware.

* 1. **Overview and Motivation:**

**Overview:**

The main motive of our project CALTALK(Voice Command Based Calculator) is to allow the users to give input by text as well as voice.

**Motivation:**

* Time consuming in given the command by typing.
* Lots of memory is wasted to install different applications for different purposes.

#### Chapter 1 Introduction

## Objective:

The aim of CALTALK is to developed a easy method of input instead of traditional way of typing on keyboard. The final application is able to perform the tasks given by the user by voice. It is useful for people with physical disabilities like handicapped because they are unable to give the input by keyboard and then they can give input by voice. It focuses on fast calculation.

## Organization of Project Report:

|  |  |
| --- | --- |
| **PHASE** | **TIME DURATION** |
| Software requirement specification | 2 weeks |
| System design | 3 weeks |
| Coding | 5 weeks |
| Testing | 2 weeks |
| Documentation | 2 weeks |
| Implementation | 1 weeks |

**SOFTWARE REQUIREMENT ANALYSIS**

System Analysis is a detailed study of the various operations performed by a system and their relationship within and outside the system .It is a systematic technique that defines goals and objectives the goal of the development is to deliver the system in the line with the user’s requirements, and analysis is this process.

System study has been conducted with the following objectives in mind: -

* + - Identify the client’s need.
    - Evaluate the system concept for feasibility.
    - Perform economical and technical analysis.
    - Allocate functional to hardware, software, people, database and other system elements
    - Establish cost and schedule constraints.
    - Both hardware and software expertise is required to successfully attain the objectives.

## Requirement Analysis

Information gathering is usually the first phase of the software development project. The purpose of this phase is to identify and document the exact requirements for the system. The user’s request identifies the need for a new information system and on investigation re-defined the new problem to be based on MIS, which supports management. The objective is to determine whether the request is valid and feasible before a recommendation is made to build a new or existing manual system continue

The major steps are –

* + - Defining the user requirements.
    - Studying the present system to verify the problem.
    - Defining the performance expected by the candidate to use requirement

#### Hardware Requirements

Processor : Intel Dual Core or More Processor

Speed : 2 GHZ

RAM : 1 GB

Hard Disk : N/A

#### Tools and Technology Tools:

* + - * Windows-10
      * Python IDE/IDLE
      * Pycharm

#### Technology:

* **Python** is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

## Feasibility Study

Feasibility study is the process of determination of whether or not a project is worth doing. Feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. I have taken a fixed time in feasibility study with my co-developer. The contents and recommendations of this feasibility study helped us as a sound basis for deciding how to precede the project. It helped in taking decisions such as which software to use, hardware combinations, etc.

#### Technical feasibility:

This is concerned with specifying equipment of software and hardware that will successfully satisfy the user requirements. The technical needs of the system may vary considerably, but might include:

* The facility to produce output in a given time.
* Response time under certain condition.
* Ability to produce a certain volume of transaction at a particular speed.
* In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system requirements. What speeds of input and output should be achieved at particular quality of printing.

According to the definition of technical feasibility the compatibility between front-end and back-end is very important. In our project the compatibility of both is very good. The speed of output is very good when we enter the data and click button then the response time is very fast and give result very quick. I never find difficulty when we use complex expression.

We use Python 3.6. The designing of GUI as well for internal working of the application:

* + Easy implementation of code.
  + Well define interface and Modules.
  + User friendly UI

At present scenario the no of Python versions are available but we have selected Python 3.6 because of the following number of reasons.

* + - It supports Google’s Speech Recognition System.
    - Easy to implement.
    - Vast collection of APIs

With the help of above support we remove defect of existing software. In future we can easily switch over any platform. To ensure that system does not halt in case of undesired situation or events. Problem effect of any module does not affect any module of the system. A change of hardware does not produce problem.

#### Operational Feasibility:

It is mainly related to human organizational and political aspects. The points to be considered are:

* + - * What changes will be brought with the system?
      * What organization structures are distributed structures are distributed.
      * What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

At present stage all the work is done manually. So, throughput and response time is too much. Major problem is lack of security check that should have been applied. Finding out the detail regarding user’s request was very difficult, because data store was in different registers and different places. In case of any problem, no one can solve the problem until the person responsible is not present.

Current communication is entirely on telephonic conversation or personal meetings. Post computerization staff can interact using internet.

Now, we will explain the last point of operational feasibility i.e. handling and keeping of software , at every point of designing I will take care that menu options are not too complex and can be easily learned and required least amount of technical skills as operators are going to be from non-computers back ground.

#### Economic feasibility:

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis: the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with cost. If benefits outweigh cost, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

At present Company have ten systems with following configuration:

* + - * Ram 4 GB or above for fast execution and reliability
      * MOTHER Board x64 based PC
      * Color Monitor 14” and 17”
      * Hard Disk 100 GB
      * Hence the economic feasibility is very good.

## Analysis

System analysis is the first step towards the software building process. The purpose of system analysis is to understand the system requirements, identify the data, functional and behavioral requirements and building the models of the system for better understanding of the system.

In the process of system analysis one should first understand that, what the present system is how it works (i.e. processes). After analyzing these points we become able to identify the problems in the present system. Upon evaluating current problems and desired information (input and output to the system), the analyst looks towards one or more solutions. To begin with, the data objects, processing functions, and behavior of the system are defined in detail.

After this models, from three different aspects of the system-data, function and behavior. The models created during the system analysis process helps in better understanding of data and control flow, functional processing, operational behavioral and information content.

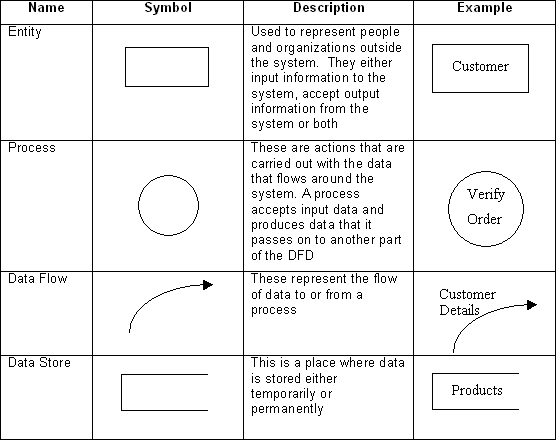
# SOFTWARE DESIGN

A software design document (SDD) is a written description of a [software](http://en.wikipedia.org/wiki/Software) product, that a software designer writes in order to give a [software development](http://en.wikipedia.org/wiki/Software_development) team overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design.

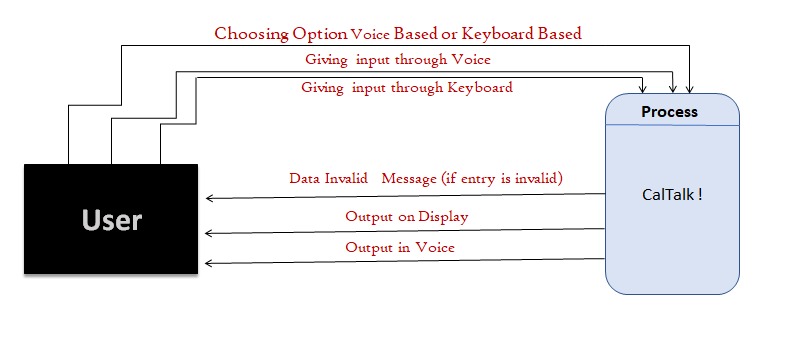
Practically, a design document is required to coordinate a large team under a single vision. A design document needs to be a stable reference, outlining all parts of the software and how they will work. The document is commanded to give a fairly complete description, while maintaining a high-level view of the software.

There are two kinds of design documents called HLDD (high-level design document) and LLDD (low-level design document).

The SDD contains the following documents:

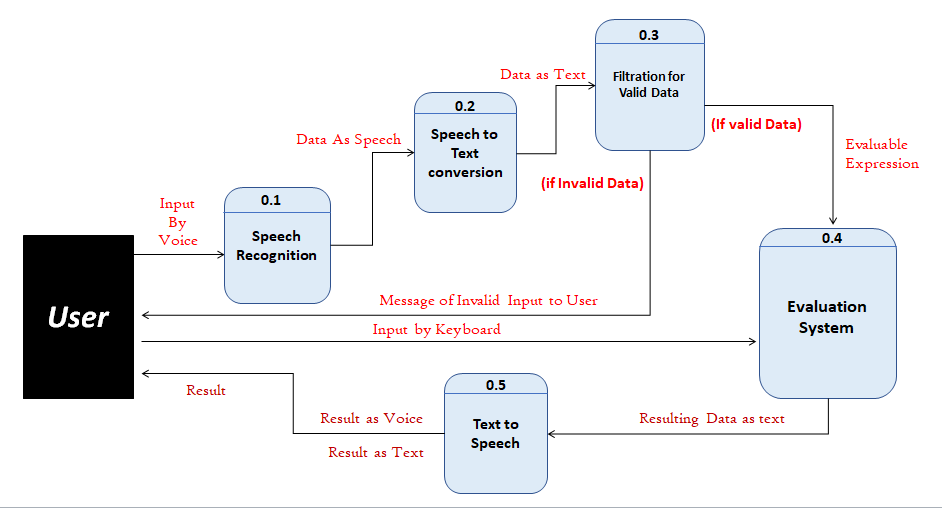
* 1. The [**data design**](http://en.wikipedia.org/wiki/Data-driven_design) describes structures that reside within the software. Attributes and relationships between [data objects](http://en.wikipedia.org/wiki/Data_object) dictate the choice of [data structures](http://en.wikipedia.org/wiki/Data_structures).
  2. The [**architecture design**](http://en.wikipedia.org/wiki/Software_architecture)uses information flowing characteristics, and maps them into the program structure. The transformation mapping method is applied to exhibit distinct boundaries between incoming and outgoing data. The data flow diagrams allocate control input, processing and output along three separate modules.
  3. The [**interface design**](http://en.wikipedia.org/wiki/Interface_design) describes internal and external program interfaces, as well as the design of human interface. Internal and external interface designs are based on the information obtained from the analysis model.
  4. The [**procedural design**](http://en.wikipedia.org/wiki/Procedural_design)describes structured programming concepts using graphical, tabular and textual notations. These design mediums enable the designer to represent procedural detail that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineering worked.

**3.1.0 DFD LEVEL 0**

****

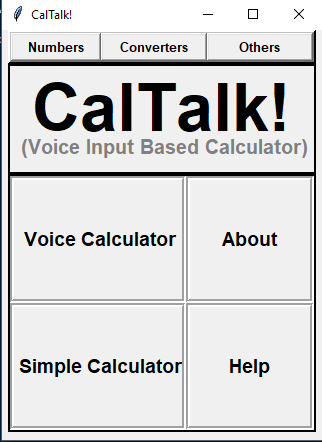
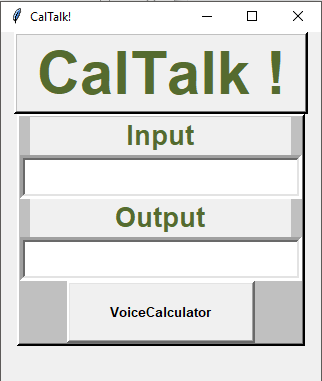
**Fig 3.2: 0 Level DFD**

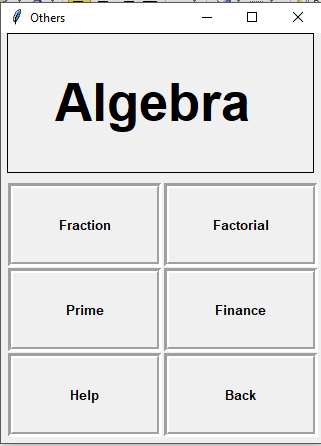
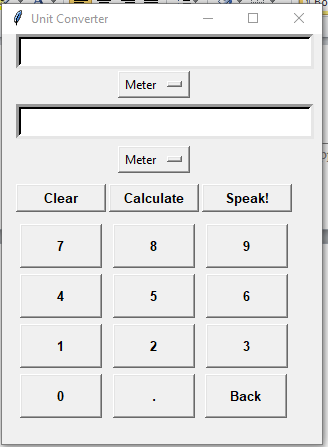
* + 1. **DFD LEVEL 1**

****

**Fig 3.3: 1 Level DFD**

**IMPLEMENTATION & USER INTERFACE**

** **

**Chapter 5**

**Software Testing**

**SOFTWARE TESTING**

## Testing

* + - Software testing is the process of executing a program with intension of finding errors in the code. It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not.
    - Generally, no system is perfect due to communication problems between user and developer, time constraints, or conceptual mistakes by developer.
    - To purpose of system testing is to check and find out these errors or faults as early as possible so losses due to it can be saved.
    - Testing is the fundamental process of software success.
    - Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases i.e. design development and maintenance phase.
    - Testing is used to show incorrectness and considered to success when an error is detected.

## Objectives of Software Testing

* + - **Software Quality Improvement:** The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So a great consideration is required for checking for quality of software.

#### Verification And Validation:

* + - * Verification means to test that we are building the product in right way .i.e. are we using the correct procedure for the development of software so that it can meet the user requirements.

16

* + - * Validation means to check whether we are building the right product or not.
    - **Software Reliability Estimation:** The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software reliability

## Principles of Software Testing

* + - All tests should be traceable to end user requirements.
    - Tests should be planned long before testing begins
    - Testing should begin on a small scale and progress towards testing in large
    - To be most effective testing should be conducted by an independent third party

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

* + - White box testing.
    - Black box testing.

#### White-box testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

17

**Chapter 5**

**Software Testing**

#### Block-box testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

## Testing fundamentals

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

## Testing Information flow:

Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code.

Test configuration includes test plan and test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied and debugging commences.

18

**Conclusion**

**CONCLUSION**

This was the first considerably large and important project undertaken by me during my BCA course. It was an experience that changed the way I perceived project development. The coding could not be started before the whole system was completely finalized. Even then there were so many changes required and the coding needed to be changed. I attribute this to inadequate information gathering from the user. Though there were many meetings with the user and most of the requirements were gathered, a few misinterpretations of the requirements still crept in. It made me realize how important the systems analysis phase is. The project is a classic example, that learning of concepts needs to be supplemented with application of that knowledge.

On the whole it was a wonderful experience developing **CALTALK** and I would have considered my education incomplete without undertaking such a project which allowed me to apply all that I have learnt and tried to develop a project that can be useful for people related to study field to calculate more easily and effectively. It is developed using PYTHON so that it can be accessed very easily and at any time. The system will be capable of perform any type of calculation like scientific and different conversions. The system is developed with an aim of usability so that it is an easy to use system that requires the least amount of user input possible. For using this system general computer knowledge is enough.

19

**BIBLIOGRAPHY & REFERENCES**

To develop this desktop application CALTALK, we used PYTHON 3.6. We take some knowledge towards GUI system from some books that are given below:

## *Python GUI Programming with Tkinter :Develop Responsive*

## *And Powerful GUI Applications with Tkinter – by Alan Forbes*

## *Core Python Programming – by Dr. R. Nageshwar Rao*

### References:

### [1]. www.w3schools.com/python

### [2]. www.stackoverflow.com

### [3]. www.geeksforgeeks.com

### [4]. www.w3schools.com/tkinter

### [5].www.tutorialspoint.com/tkinter

20