Project Report
On
Andha-dhun

A Gesture and Speech Based Computer Control Application

Bachelor of Technology
Computer Science and Technology

Submitted by :-

Urvi Rawat (17070122070)

Vidisha Wagle (17070122075)

Yagnik Bhargav (17070122077)

Yash Dave (17070122078)

Under the Guidance of

PROF. POOJA V. KAMAT

SYMBIOSIS INSTITUTE OF TECHNOLOGY

(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)

Pune - 412115

Table of Content

1 . INTRODUCTION 2

1.1. PURPOSE

1.2. FEASIBILITY STUDY

1.3. INTENDED AUDIENCE

1.4. PRODUCT SCOPE

1.5. REFERENCES

2 . OVERALL DESCRIPTION 3

2.1. PRODUCT PERSPECTIVE

2.2. PRODUCT FUNCTIONS

2.3. TECHNOLOGY

2.4. CONSTRAINTS

2.5. PROCESS MODEL

2.6. ASSUMPTIONS AND DEPENDENCIES

3 . FUNCTIONAL REQUIREMENTS 5

3.1. GESTURE CONTROLLED MOUSE

3.2. GESTURE CONTROLLED KEYBOARD

3.3. TEXT TO SPEECH

3.4. SPEECH TO TEXT

3.5. CALIBRATION

3.6. SPECIAL SCREEN ADJUSTMENTS

4 . NON-FUNCTIONAL REQUIREMENTS 8

4.1. PERFORMANCE REQUIREMENTS

4.2. SECURITY REQUIREMENTS
4.3. SAFETY REQUIREMENTS
4.4. SOFTWARE QUALITY ATTRIBUTES

5 . PROJECT SCHEDULING AND MANAGEMENT 9

5.1. FUNCTIONAL DECOMPOSITION
5.2. ROLES AND RESPONSIBILITIES
5.3. TASK DURATION
5.4. BUDGET ESTIMATION
5.5. RMMM

1. Introduction

5.1 1.1 Purpose

The purpose of this document is to enumerate the requirements, procedures and applications of this project. We aim to analyze all requirements, ranging from, technologies to be used, market analysis, dependencies, limitations and so on. Also, provide a detailed study of all the functions that our project would perform. Finally our aim is to scrutinize every application that our endeavor would offer.

5.2 1.2 Feasibility study

On an average, 17% of the total cost of a PC setup is spent on secondary devices like a keyboard or a mouse. Our application aims to completely **eliminate this overhead expenditure.** The visually impaired can use their computers by installing hardware or software that simulates the human voice reading the computer screen or produces hard-copy output braille. However, these are present as different hardware and software, our project aims to provide **one single platform** for all these services. One part of our project focuses on screen magnifications adjustments for the borderline blind.

5.3 1.3 Intended audience

Our project targets 3 sections of the society:

- People who want to reduce the cost of their PC setup
- The Visually Impaired
- People with Visual Disorders or Color Blind

5.4 1.4 Product scope

This application aims to make computers independent of secondary devices like the keyboard and mouse using concepts of Computer Vision. Along with this, it makes computers usable for the visually impaired using Gesture control, text to speech and speech to text technologies. The ultimate motive is to provide a better experience for our intended audience.
Our project will only be implemented as a software and will have no extra hardware needed.

5.5 1.5 References

- Python Documentation for OpenCV, Python Auto GUI, and other packages
- YouTube channels – Sentdex()and The NewBoston()
- Stackoverflow-www.stackoverflow.co

2. Overall Description

2.1 Product Perspective

Our project is a software that will use gesture controls and speech-text conversions to help the user navigate and use the PC. It would help users to use less efforts and resources required to control the pointer of the mouse or any other accessibility options. It would reduce human effort and would provide better substitutes of mouse and keyboard.

2.2 Product Functions

Our project would provide functionalities like Gesture control for mouse as well as keyboard, speech-to-text , text-to-speech and also some features like Color adjustments for the accessibility of the users.

2.3 User classes and characteristics

2.3.1 Normal User -The normal user will access the gesture based functions that will replace the need of keyboard and mouse. He can also use the speech to text features.

2.3.2 Visually Impaired – The visually impaired can use the text to speech to get what is there is on the screen and the speech to text to perform various functions.

2.3.3 People with Visual Disorders – The screen settings can be optimized as per user needs.

2.4 Technology

The project will be implemented in python and will make use of speech to text conversion libraries (Speech Recognition using google speech API) and computer vision libraries (OpenCV)

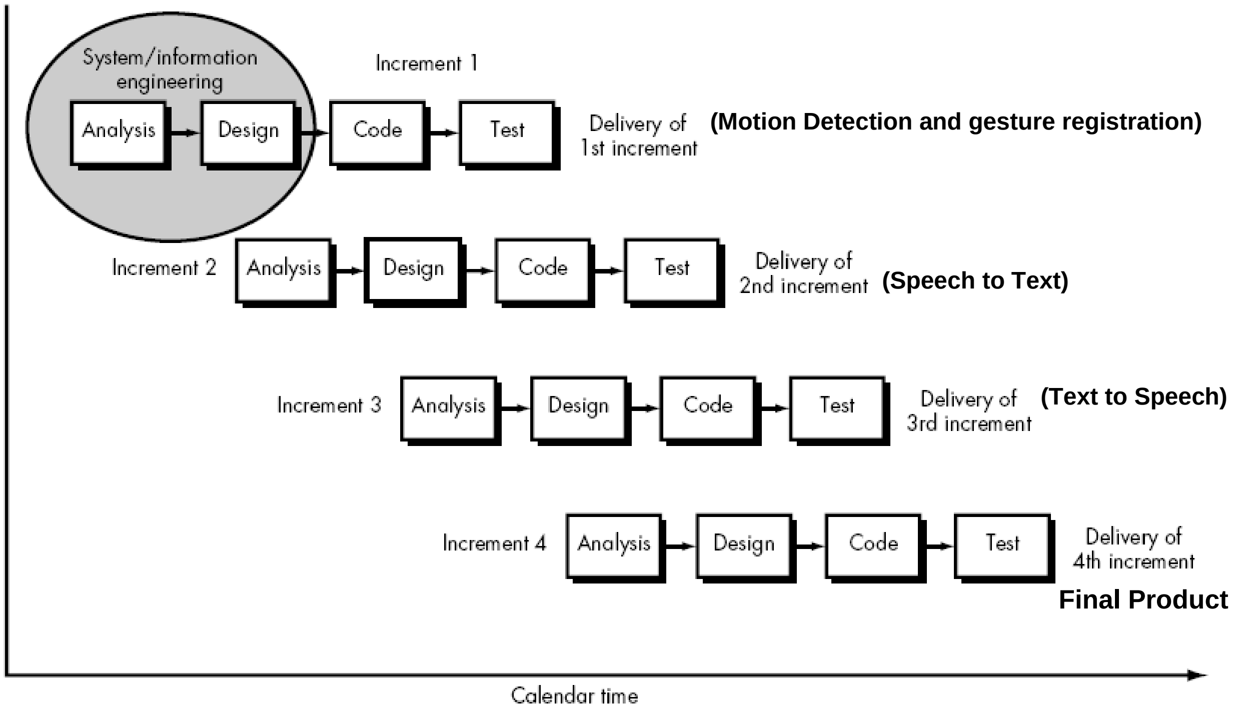
2.5 Constraints

1.The project will only be beneficial to the people who use PC setups.
2.The text-speech conversions will be limited to English language only.
3.Screen adjustments are subject to monitor driver permissions
4.The gestures will be controlled by a black dot on the users hands.

2.6 Process Model

Our project will make use of an incremental model.

This model will be most appropriate for our project since every aspect of our project will be divided into phases and feedback will be taken at every stage. We would want to divide all our modules into 4 basic stages namely - analysis, design, coding and testing.



2.7 Assumptions

1.The user has a good quality webcam to detect motions and gestures.

2.The user has good quality working microphone and speakers to effective text-speech conversions

3.The user uses a black dot which is clearly visible.

4.The user is capable enough for setting the application up according to his/her needs

2.8 Dependencies

1.Python 3.6 and above must be running on the PC.

2.All the required libraries must be installed before using the applications.

3. Functional Requirements

3.1. Gesture controlled Mouse

3.1.1 Description and Priority

The module provides a purely virtual mouse controlled by the user's finger. Gestures are used in order to perform operations like clicking and scrolling. We use the feed from the user's webcam in order to do so This module has high priority.

3.1.2 Cursor movement via finger detection and tracking

Input: Consecutive real time frames from the webcam, consisting of the finger with a black dot in order to track finger movement with ease.
Output: Movement of the pointer on screen wrt the movements of the user's finger.

3.1.3 Mouse Click using gesture recognition

Input: Consecutive real time frames from the webcam, in order to recognize gestures, if any.

Output: Performing right click, left click, drag & drop and certain other tasks like pulling up the keyboard, or launching the video player etc.

3.2. Gesture controlled on-screen Keyboard

3.2.1 Description and Priority

An on-screen virtual keyboard is displayed. The user must point at the keys for a few seconds in order to type. The feed from their webcam is used in order to accomplish this task.
This module has high priority.

Input: user points at keys displayed on the on-screen keyboard, pausing for a few seconds to denote a keypress.
Output: A marker appears on the pressed key. Also, the characters corresponding to the key appears in the text editor.

3.3 Text-to-Speech Converter

3.3.1 Description and Priority

The text-to-speech converter is meant to process the input ‘text’ and convert it into an audio file. The language processed by this module is English only. The module has a medium priority and is mainly added for the convenience of visually impaired people.

Input: The user will upload the ‘text document’/’text’ to be translated into an audio file. The file must in English language only.
Output: The user gets the mp3 audio file in English which will be automatically played in the background.

3.4 Speech to text converter

3.4.1 Description and Priority

The speech-to-text converter is meant to process the input ‘speech’ and convert it into a text file. The language processed by this module is English only. The module has a medium priority.

Input: Audio input in .mp3 or .wav file or the microphone.

Output: Generated text.

3.5 Calibration

3.5.1 Description and Priority

When the user installs the software, the computer needs to understand the user movements and hence needs to calibrate as per the screen resolutions and the hand gestures as they vary from person to person

Input: User pointed Coordinates

Output: Calibration values in X and Y coordinates

3.6 Special screen adjustments

3.6.1 Description and Priority

Targeted towards the borderline blind audience, this module helps the user tweak the screen settings such as color scheme, icon size etc., making it easier for those with special visual impairments to interact with the PC.
This module has low priority.

Input: The screen with default settings
Output: Customized screen by the user (mainly color changes)

4. Other Non-Functional Requirements

4.1. Performance Requirements

The purpose of the software is to replace the mouse and keyboard for cheaper, more convenient navigation; hence it is crucial for the software to respond to user’s gesture and voice as quickly as possible for a seamless experience. The tracking of the user’s finger and simultaneous movement. The estimation of user’s posture, comparison with reference pose and display of accuracy score, angle correction must happen as soon as the user emulates the pose depicted by the picture. Specifically, the response time must not exceed 2 seconds.

4.2. Security Requirements

The major security concern for any user is proper login in the account. For this purpose, proper login mechanism should be used to avoid any form of hacking. Therefore, security is provided from the unwanted use of the software. Also user may be concerned on allowing the camera as this is a sensitive data, but we have considered it and we shall not save any of this data for our development purposes or other purposes and hence user may feel safe for the privacy he gets on our software.

4.3. Safety Requirements

The transmission of information should be done securely and transmitted to the server without any change in the entered information. And as stated the user data is completely on the computer only for an instance and hence the user should not be concerned of his digital safety.

4.4. Software Quality Attributes

4.4.1 Availability

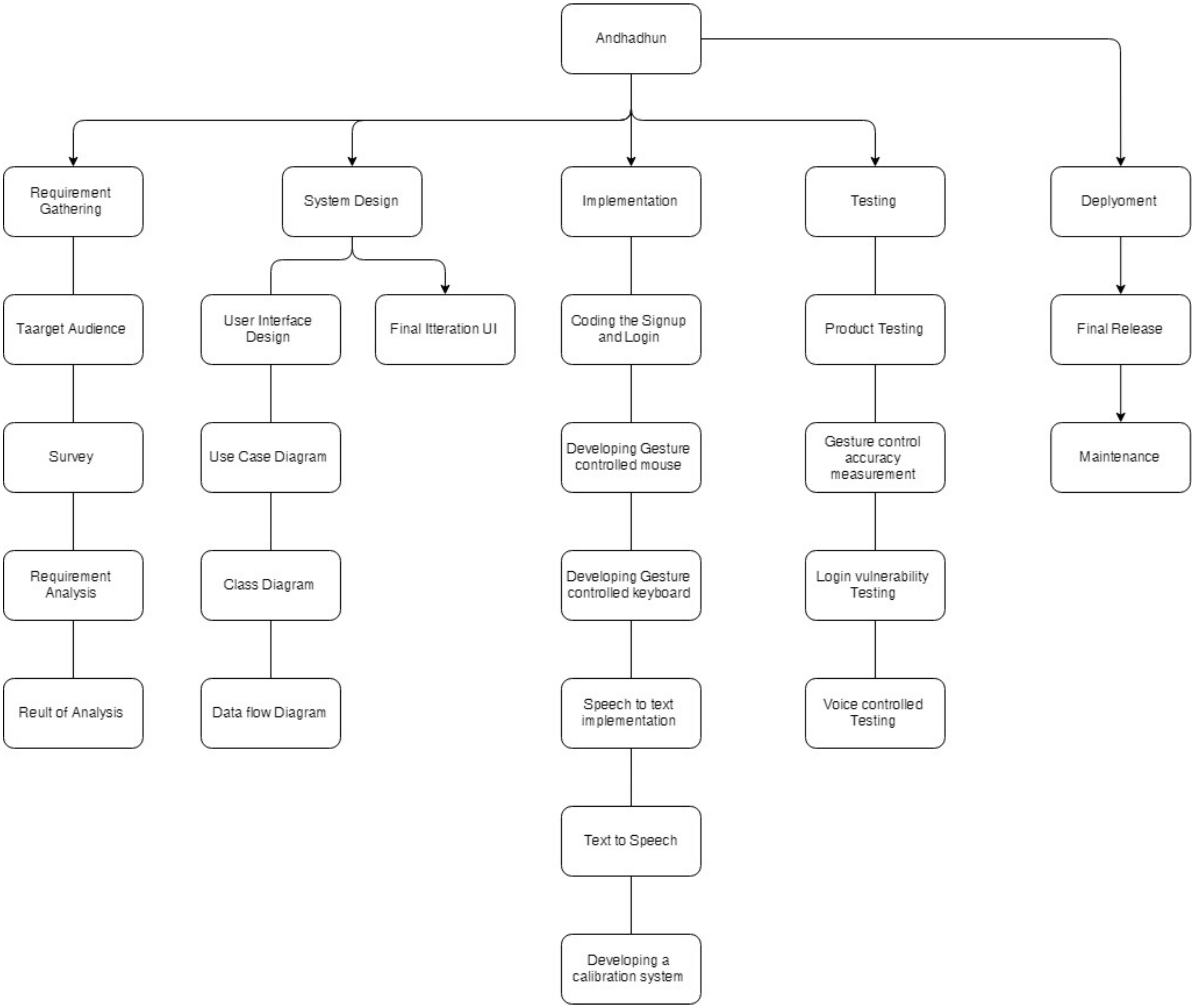
It means that the system always has something to function and in case of any type of failure an error messages should pop up. An error message appears when something goes wrong to avoid availability problems.

4.4.1 Usability

It means checking that the system is easy to handle and navigation through software occurs in the most expected way without any delays.

5. Project Scheduling and Estimation

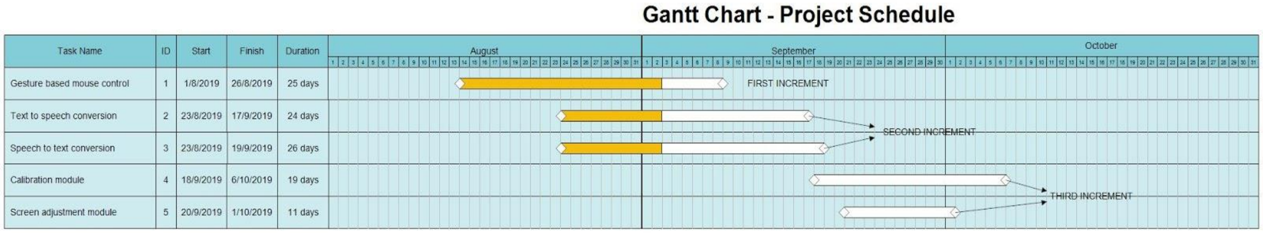
5.1. Functional Decomposition



5.2. Roles and Responsibilities

Sr. no	Task	Roles and Responsibilities
1.	Requirement Gathering	Urvi Rawat Vidisha Wagle, Yash Dave, Bhargav Yagnik
2.	Planning and Software Requirement	
3.	Project Scope and Documentation	
4.	UI Design	Bhargav Yagnik, Urvi Rawat
	Use Case Diagram and Class Diagram	Yash Dave , Vidisha Wagle
5.	Login	Urvi Rawat and Vidisha Wagle
	Developing Gesture based control of mouse	Urvi Rawat and Bhargav Yagnik
	Developing Gesture based control of Keyboard	Urvi Rawat and Bhargav Yagnik
	Speech to Text	Yash Dave
	Text to Speech	Vidisha Wagle
	Calibration	Yash Dave and Bhargav Yagnik
6.	Testing- Gesture Control, Login Vulnerabilities, Speech Testing	Urvi Rawat Vidisha Wagle, Yash Dave, Bhargav Yagnik
7.	Deployment	

5.3. Task Duration



Gantt Chart - Project Schedule

Task Name	ID	Start	Finish	Duration
Gesture based mouse control	1	1/8/2019	26/8/2019	25 days
Text to speech conversion	2	23/8/2019	17/9/2019	24 days
Speech to text conversion	3	23/8/2019	19/9/2019	26 days
Calibration module	4	18/9/2019	6/10/2019	19 days
Screen adjustment module	5	20/9/2019	1/10/2019	11 days

5.4. Budget Estimation

The estimation of the effort and development time for the project is based upon Basic COCOMO (Cost Constructive Model). The basic COCOMO estimation model is given by the following expressions:

Effort = a_1 * (KLOC) ^{a_2} PM

T_{dev} = b_1 * (Effort) ^{b_2} Months

We would be using the Semi detached type of project for considering the values of a_1,a_2,b_1,b_2 . The major motive of selecting the semi-detached as our team has a limited experience of doing projects of this type. There are areas like gesture recognition etc. are new to our team while we also have experience using Computer Vision and speech recognition.

Hence a_1=3.0, a_2=1.12, b_1=2.5, b_2=0.35.

Line of Code: There are 5 main functionalities in the code, and it should take an average of 220 lines of code and hence the estimated line of code is around 1.1k lines. + 100 lines for the front-end.

Hence Lines of Code =1.2kLOC

Effort = 3.0 * (1.2) $^{1.12}$ = 3.67 PM

T_{dev} = 2.5 * (3.67) $^{0.35}$ Months = 3.94 Months

So,

Effort = 3.67 PM

T_{dev} = 3.94 Months

5.5. RMMM

No.	Risk	Category	Probability	Impact
1	Gestures unrecognized	Technical	30%	Catastrophic
2	Screen adjustments incorrect	Technical	10%	Catastrophic
3	Accents too heavy to understand	Customer related	30%	Marginal
4	Poor Quality Documentation	Process	20%	Marginal
5	Incorrect process model choice	Process	10%	Catastrophic
6	Poorly commented code	Process	40%	Negligible

Risk:

- **Gestures unrecognized**
  - Mitigation**:**
    If a gesture is not recognized, it may result in actions the user doesn't intend to take. There is a risk that the user may want to perform some action but instead all the user data may be deleted. This will cause harm to the image of the product and is hence classified a Catastrophic risk.
  - Monitoring:
    When making the module, the developer must always be aware and sure about the classification dataset used so that no such risks take place.
  - Management:
    In case such an issue occurs, the developer team must include a "KILL-SWITCH" which will help not cause any further damage to the PC.

- **Screen adjustments incorrect**
  - Mitigation:
    If the screen adjustments are incorrect, it could cause even more trouble to the user and it may be difficult to even go back to default settings. It could cause the user extra effort and possibly pain to reset it. Hence it is classified as a catastrophic risk.
  - Monitoring:

When making the module, the developer must always be sure about the users' choices and preferences.

- Management:
  In case such an issue occurs, the developer team must include a default setting mode which can be used by all users.

- **Accents too heavy to understand**

  - Mitigation:
    Sometimes certain people have thick accents which could cause some errors speech recognition. These errors could result in actions which were not intended to be performed.

  - Monitoring:
    When making the module, the developer must train the datasets well and use rich datasets for training.

  - Management:
    In case such an issue occurs, the user should be asked to calibrate their voice again.

- **Poor Quality Documentation**

  - Mitigation
    In order to prevent this from happening, members who are in charge of developing the documentation will keep in contact with each developer on the team. Meetings will be held routinely to offer documentation suggestions and topics. Any topic deemed missing by a particular developer will be discussed and it will be decided whether or not to add that particular topic to the documentation. In addition, beta testers will be questioned about their opinion of the documentation.

  - Monitoring
    Throughout development or normal in and out of house testing, the development team and or beta testers will need to keep their eyes open for any possible documentation topics that have not been included.

  - Management
    Should this occur, the organization would call a meeting and discuss the addition of new topics, or removal of unnecessary topics into the documentation.

- **Incorrect process model choice**
  - Mitigation:
    An incorrect process model can lead to huge wastage of time. The team should study the SRS document carefully before making a choice.

  - Monitoring:
    The developers must make sure every aspect has to be crystal clear. Also, at every checkpoint, the team has to make sure the process model fits.

  - Management:
    In case such an issue occurs, the team will have to work overtime to recover from the damages caused. There will be huge wastage of time.

- **Poorly commented code**

  - Mitigation Poor code commenting can be minimized if commenting standards are better expressed. While standards have been discussed informally, no formal standard yet exists. A formal written standard must be established to ensure quality of comments in all code.

  - Monitoring Reviews of code, with special attention given to comments will determine if they are up to standard. This must be done frequently enough to control comment quality. If they are not done comment quality could drop, resulting in code that is difficult to maintain and update.

  - Management Should code comment quality begin to drop, time must be made available to bring comments up to standard. Careful monitoring will minimize the impact of poor commenting. Any problems are resolved by adding and refining comments as necessary.