VIRTUAL PIANO

Capstone Project Report

End-Semester Evaluation

Submitted by:

(101683035) Akshay Sharma

(101503008) Abhishek Sharma

BE Fourth Year, CSE

CPG No: 84

Under the Mentorship of

Dr Singara Singh

Associate Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

TIET, Patiala

October 2018

ABSTRACT

Pianos are large instruments that cannot be carried everywhere. Even electric pianos need to be carried in a large bag and are prone to damage in travelling. So here we propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it. We use a raspberry pi attached to a camera along with the plastic sheet to make a virtual piano. We use image processing to divide the plastic sheet into sections and assign particular tomes to it. We the detect human fingers through the plastic sheet and simulate associate piano tone for each section to play piano tones using a speaker. Thus, we provide a virtual piano which is actually a transparent light weight plastic board that can be carried around roughly.

DECLARATION

We hereby declare that the design principles and working prototype model of the project entitled Virtual Piano using Raspberry pi is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of Dr Singara Singh during 6th and 7th semester (2018).

Date: 09/10/2018

Roll No.	Name	Signature
101683035	Akshay Sharma	
101503008	Abhishek Sharma	

Counter Signed By:

Faculty Mentor:

Dr. Singara Singh

Associate Professor

Computer Science & Engineering Department,

TIET, Patiala

ACKNOWLEDGEMENT

We would like to express our thanks to our mentor Dr Singara Singh. He has been of great help in our venture, and an indispensable resource of technical knowledge. He is truly an amazing mentor to have.

We are also thankful to Dr Maninder Singh, Head, Computer Science and Engineering Department, entire faculty and staff of Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards successful completion of this project. We thank all those who have contributed either directly or indirectly towards this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement. They always wanted the best for us and we admire their determination and sacrifice.

Date: 09/10/2018

Roll No.	**Name**	**Signature**
101683035	Akshay Sharma	
101503008	Abhishek Sharma	

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

HGR	Hand Gesture Recognition
RGB	Red Green Blue
AR	Augmented Reality
FAQ	Frequently Asked Questions
HSV	Hue Saturation Value
TIET	Thapar Institute of Engineering & Technology
SD	Secure Digital Card
WBS	Work Breakdown Structure
DFD	Data Flow Diagram
BLOB	Binary Linked Object
LCD	Liquid Crystal Display
SDK	Software Development Kit
API	Application Program Interface

INTRODUCTION

1.1 Project Overview

1.1.1. Technical Terminology

Pianos are large instruments that cannot be carried everywhere. Even electric pianos need to be carried in a large bag and are prone to damage in travelling. So here we propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it. We use a raspberry pi attached to a camera along with the plastic sheet to make a virtual piano. We use image processing to divide the plastic sheet into sections and assign particular tomes to it. We the detect human fingers through the plastic sheet and simulate associate piano tone for each section to play piano tones using a speaker. Thus, we provide a virtual piano which is actually a transparent light weight plastic board that can be carried around roughly.

Hand Gesture Recognition (HGR) is a very popular and effective way used for the human machine communication. It has been used in many applications including embedded systems, vision-based systems and medical applications. In HGR, fingertip detection is an important part if image base models are being used. HGR systems face many problems in skin segmentation due to luminance and intensity in images [1]. The fingertips detection models mostly have assumption about the hand direction; this restricts the natural expression of humans. Processing time is another key factor in image-based processing algorithms.

Here we are focusing on direction invariant fingertip detection of natural hand with real time performance. This work requires no glove, sensors or color strips to detect the fingertips. The only assumption is that user will show the hand to system, facing the palm to the system while the direction of hand is not restricted. User is free to show hand in any direction as naturally hands move [11]. This report also presents a figure cropping method based on hand size, which will fasten the further process as the processing pixels would be reduced after cropping.

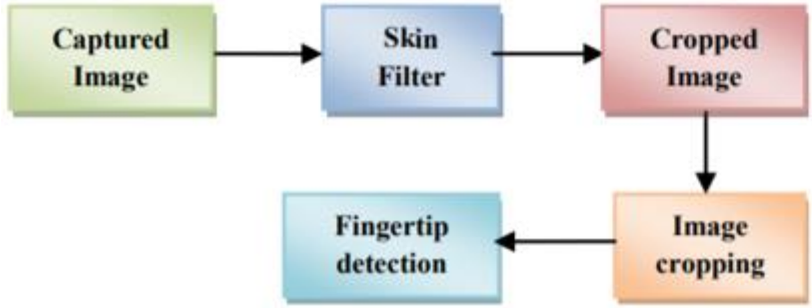

Figure 1.1: Flowchart for depicting Fingertip detection

A lot of work has been done in this area for dynamic hand gesture recognition using fingertip detection. A survey on fingertip-based methods could be found out in. There are several limitations in existing approaches. Garg used 3D images in his method to recognize the hand gesture, but this process is complex and also not time efficient. The Processing time is one of the very critical factors in real time applications. Aznaveh presented an RGB vector-based method for skin detection in images. Yang analyses the hand contour to select fingertip candidates, then finds peaks in their spatial distribution and checks local variance to locate fingertips [6]. This method is not invariant to the orientation of the hand. There are other methods, which are using directionally Variant templates to detect fingertips. Few other methods are dependent on specialized instruments and setup like the use of infrared camera, stereo camera, a fixed background or use of markers on hand. This report describes a novel method of motion patterns recognition generated by the hand without using any kind of sensor or marker.

The fingertips detection should be near to real time if it is going to process video. Generally, image-based models work on pixel by pixel and do hand Segmentation and work only on region of interest [6]. However, most hand segmentation methods can’t do a clearly hand segmentation under some conditions like fast hand motion, cluttered background, poor light condition. If the hand segmentation is not valid, then detection of fingertips can be questionable. Researchers used infrared camera to get a reliable segmentation [10]. Few researchers in their work limit the degree of the background clutter, finger motion speed or light conditions to get a reliable segmentation. Raheja and few others also used 3D mapping using specialized device like KINECT for hand segmentation [2]. Some of fingertip detection methods can't localize accurately multidirectional fingertips. Researchers assumed that the hand is always pointing upward to get precise localization.

Deaker proposed a piano tutoring system which would test the user's knowledge by asking them to play particular notes or chords on a paper keyboard with a printed design [4]. Deaker's proposal is similar to ours in that it requires the detection of the printed keyboard's individual keys in order to determine if the user has followed instructions correctly. However, Deaker's design has several limitations, perhaps the most significant of which being that the identification of keys is not entirely achieved by computer vision alone. A particular number of keys are required to be visible to the camera during calibration (14 white keys, starting from a C key and ending on the B key just under two octaves above) and the software assumes that this requirement has been met.

Another computer vision based piano tutor was proposed by Dawe [6]. In this case, the system is intended for relaying a student's choice of fingering to their teacher during a remote piano lesson. Similarly, to Deaker's proposal, Dawe's method requires a calibration stage where the user is required to select a region of interest – in order to narrow the video frames down to the area occupied by the instrument's keyboard. This brings with it the same limitations imposed by Deaker's method; that is the camera must remain stationary at all times, and the keyboard must not be obscured during the calibration process.

To find the key locations during the calibration phase, Dawe used horizontal and vertical edge detection to determine the boundaries of the keyboard and its individual keys. Unfortunately, a major limitation of Dawe's method was that a translation between these detected edges and a description of the location of each key of the keyboard was unable to be implemented.

After examining these proposed methods, it was clear that a satisfactory method of keyboard and key detection had not been found. The following works were more successful in their attempts to detect individual piano keys.

The work by Barakonyi and Schmaltzier demonstrates a variety of 3D augmented reality (AR) applications, including the Augmented Piano Tutor application [2]. The application displays an AR overlay on the user’s electronic keyboard, indicating which keys are to be played.

The method proposed by Gorodnichy and Yogeswaran attempts to solve the same problem as identified by Dawe; that is, Gorodnichy and Yogeswaran present a solution to allow current remote piano teaching systems to communicate fingering information [4]. There is one major limitation of Gorodnichy and Yogeswaran’s method; the camera used is not an off-the-shelf webcam, but instead a more expensive video camera with high quality output and zoom functionality is required.

The final keyboard detection method we shall discuss is from a paper by Huang, Zhou, Yu, Wang and Du [3], where a marker less AR piano system is proposed. The 3D AR result is very similar to that achieved by Barak Onyi and Schmaltzier, although in this case a fiducial marker is not required. To identify the keys of the keyboard, a similar method to that of Gorodnichy and Yogeswaran is employed. After identifying the piano keys, this solution uses OpenGL to augment the video stream with 3D graphics.

This was seemed to accomplish almost all of what we were setting out to achieve. Consequently, the work by Huang, Zhou, Yu, Wang and Du is regarded as a target result for our paper, with our application intended to match the results produced by the marker less AR piano system as best as possible. More specifically, the system by Huang et al. managed the keyboard identification and rendering of the video feed, with overlays, at a frame rate of approximately 15 frames per second [6]. We hoped to match or exceed this performance with our solution.

1.1.2. Problem Statement

Pianos are large instruments that cannot be carried everywhere. Even electric pianos need to be carried in a large bag and are prone to damage in travelling. So here we propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it.

1.1.3. Goal

1. Assigning the respective tones of piano to the corresponding segments of Virtual Piano made using a semi-transparent plastic sheet.
2. Real time motion detection of fingers using a camera linked with Raspberry pi.
3. To play the tone assigned to a segment on detection of finger on that particular segment.

1.1.4. Solutions

As a final note on the system developed by Huang et al., there is a small way in which the design could arguably be improved upon: The marker less AR piano system requires a one-time calibration for each camera and/or piano used, and for the 3D registration to be possible, the dimensions of the keyboard need to be known by the software [13]. This requires that the user manually measure their piano keyboard, then input the information into the AR program. At the expense of 3D augmentations, this potential inconvenience can be removed.

1.2 Need Analysis

Piano is almost certainly the largest and most complex mechanical device. A standard piano has 230 strings and about 10,000 separate moving parts. Over the last decade, alternatively sized keyboards have become available in a very limited way. At the same time, various academics, researchers and manufacturers such as Steinbuhler & Company have started to raise awareness and discuss the possibility that the current size keyboard is not ideal for all, and that keyboards with narrower keys may well suit the majority of the population [5]. However, they encounter considerable resistance to contemplate change from the ‘one-size suits all’ mentality. A major disincentive for a pianist wanting to switch to a keyboard of a different size is that they are not available - in practical terms every piano keyboard anywhere is the standard size. Major piano manufacturers do not produce pianos with narrower keys, except perhaps as a rare, ‘special’ order. A handful of small manufacturers can and do, but not in enough supply: small-scale production, logistics and cost prohibit worldwide access. The plight of pianists who do have reduced size keyboards at home illustrates the complexities [7]. While they may reach a very high standard with challenging repertoire on the narrower keys, away from their own keyboard the only option are pianos with the standard keys – meaning that they struggle to play many pieces as well and have to drop some of their repertoire completely. To play outside their home they will need easy access to a 'standard' keyboard for practice – options include a grand piano with interchangeable keyboards or acquiring a standard keyboard in a good quality digital piano. As keyboards with narrower keys become more widely accepted and accessible in the community this problem will be reduced.

With the skyrocketing costs of new pianos over the past decade or two, the prices of used pianos have increased dramatically. Another upward pressure on the cost of a used piano is that there are fewer good used pianos available. A large portion of the big old uprights are now old enough that they have serious mechanical problems which cost more to repair than they are worth so they are being scrapped [1]. Many of the less expensive pianos built in the 60's and 70's are also coming to the end of their useful life and are being taken to the dump or scrapped. 15-20 years ago, a really good used piano could be purchased for $ 300-500 and something that held tune and played could be purchased for $ 100-200.

Today, the piano buyer needs to expect to pay a minimum of $ 1,200-1,500 for a piano that will hold tune and plays reasonably well and doesn't look horrible in your living room. Most used pianos that are for sale are on the market because the family has lost interest in playing the piano. Since there has been no interest in it, it probably has not been tuned or cared for in several years. The piano buyer needs to keep in mind that they will have to remedy that neglect once the piano is moved to their home [7]. It is possible that the piano has fallen way below standard pitch and will require a pitch raise (see FAQ "What is a Pitch Raise") to bring it back into tune. That pitch raises and 1st tuning could cost close to $ 200. It is also possible that the piano has not been cleaned or regulated in many years (it is not surprising to find that a piano has NEVER been serviced) so that is doesn't play very smoothly or responsively. Cleaning and regulating a vertical piano can cost around $ 400.

The greatest risk in purchasing a used piano is in whether it will hold tune or not. There is much less risk in buying a piano that has been tuned regularly and the service records are available. That is not a guarantee that it will continue to hold tune in the future but is certainly a good indicator. The buyer should be willing to pay significantly more for such an instrument than one that has not had any care for a number of years [3]. A piano technician can make an educated guess as to the condition of a piano in a brief visit, but without a history of working with the instrument, it is but an educated guess.

1.3 Research Gaps

This piano cannot be used to play complete songs because the piano is being played with the help of finger detection which is done using a camera and while playing a song it is necessary to move our fingers from one piano segment to other piano segment but the constraint in this piano is that when we move our fingers from one segment to other segment, the segments between these two segments are also detected by the camera and thus the tones corresponding to the segments in between is also played. So, this is a research gap as we could not find the solution of this problem in any of the research papers.

1.4 Problem Definition and Scope

Moving piano from one home to another without the utmost care can easily cause unforeseen problems. Hence Pianos are large instruments that cannot be carried everywhere. Even electric pianos need to be carried in a large bag and are prone to damage in travelling. Today, the piano buyer needs to expect to pay a minimum of $ 1,200-1,500 for a piano that will hold tune and plays reasonably well and doesn't look horrible in your living room. Most used pianos that are for sale are on the market because the family has lost interest in playing the piano.

So here we propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it. The solution mainly focuses on the cost and budget of the Music Enthusiastic. Virtual Piano will reduce the problem of carrying large sized pianos. The Scope of the project mainly focuses on the following things: -

1. Large Pianos are prone to damage in travelling and these Virtual Pianos will take care of this issue.
2. Virtual Pianos will be much cheaper than traditional pianos.
3. It will help young children to get a feel for music and will act as a stepping stone for some of the world's greatest artists.

1.5 Assumptions and Constraints

S. No.	Assumptions
1	Camera should be mounted properly.
2	The camera should be mounted in such a way such that it focusses on each segment properly.
3	The camera should be calibrated before the system starts.
4	The piano should be set up in bright lighting conditions.
5	The size of the piano will depend on the coverage area of camera.
6	The system should fulfil all the basic requirements needed for the project.
7	This piano cannot be used to play complete songs because the piano is being played with the help of finger detection which is done using a camera and while playing a song it is necessary to move our fingers from one piano segment to other piano segment but the constraint in this piano is that when we move our fingers from one segment to other segment, the segments between these two segments are also detected by the camera and thus the tones corresponding to the segments in between is also played.

Table 1.1: Assumptions and Constraints

1.6 Approved Objectives

1. Assigning the respective tones of piano to the corresponding segments of Virtual Piano made using semi-transparent plastic sheet.
2. Real time motion detection of fingers using a camera linked with Raspberry pi.
3. To play the tone assigned to a segment on detection of finger on that particular Segment.

1.7 Methodology Used

Pianos are large instruments that cannot be carried everywhere. Even electric pianos need to be carried in a large bag and are prone to damage in travelling. So here we propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it. We use a raspberry pi attached to a camera along with the plastic sheet to make a virtual piano. We use image processing to divide the plastic sheet into sections and assign particular tomes to it. We then detect human fingers through the plastic sheet and simulate associate piano tone for each section to play piano tones using a speaker. Thus, we provide a virtual piano which is actually a transparent light weight plastic board that can be carried around roughly.

We propose a method for identifying a piano keyboard present in the video footage of a standard webcam with the goal of teaching chords, scales and suggested finger positions to a beginner pianist. Our keyboard identification method makes use of binary thresholding, Sobel operators and Hough transforms, as well as proposed algorithms specific to this application, to first find an area resembling a piano keyboard before narrowing the search to detect individual keys [9]. Through the use of our method the keys of a piano keyboard were successfully identified from webcam video footage, with a tolerance to camera movement and occluded keys demonstrated. This result allowed the augmented reality style highlighting of individual keys, and the display of suggested fingering, for various chords and scales.

1.8 Project Outcomes and Deliverables

1. Large Pianos are prone to damage in travelling and these Virtual Pianos will take care of this issue.
2. Virtual Pianos will be much cheaper than traditional pianos.
3. It will help young children to get a feel for music and will act as a stepping stone for some of the world’s greatest artists.
4. Here, we are delivering a portable virtual piano that just uses semi-transparent plastic sheet that can be carried anywhere and does not have any electronic components in it.

1.9 Novelty of Work

In this project, we are focusing on direction invariant fingertip detection of natural hand with real time performance. This work requires no glove, sensors or color strips to detect the fingertips. The only assumption is that user will show the hand to system, facing the palm to the system while the direction of hand is not restricted. User is free to show hand in any direction as naturally hands move [11]. This report also presents a figure cropping method based on hand size, which will fasten the further process as the processing pixels would be reduced after cropping.

A lot of work has been done in this area for dynamic hand gesture recognition using fingertip detection. A survey on fingertip-based methods could be found out in. There are several limitations in existing approaches. Garg used 3D images in his method to recognize the hand gesture, but this process is complex and also not time efficient. The Processing time is one of the very critical factors in real time applications. Aznaveh presented an RGB vector-based method for skin detection in images. Yang analyses the hand contour to select fingertip candidates, then finds peaks in their spatial distribution and checks local variance to locate fingertips [6]. This method is not invariant to the orientation of the hand. There are other methods, which are using directionally Variant templates to detect fingertips. Few other methods are dependent on specialized instruments and setup like the use of infrared camera, stereo camera, a fixed background or use of markers on hand. This report describes a novel method of motion patterns recognition generated by the hand without using any kind of sensor or marker.

The fingertips detection should be near to real time if it is going to process video. Generally, image-based models work on pixel by pixel and do hand Segmentation and work only on region of interest [6]. However, most hand segmentation methods can't do a clearly hand segmentation under some conditions like fast hand motion, cluttered background, poor light condition. If the hand segmentation is not valid, then detection of fingertips can be questionable. Researchers used infrared camera to get a reliable segmentation [10]. Few researchers in their work limit the degree of the background clutter, finger motion speed or light conditions to get a reliable segmentation. Raheja and few others also used 3D mapping using specialized device like KINECT for hand segmentation [2]. Some of fingertip detection methods can't localize accurately multidirectional fingertips.

2.1 Literature Survey

2.1.1 Theory Associated with Problem Area

A lot of work has been done in this area for dynamic hand gesture recognition using fingertip detection. A survey on fingertip-based methods could be found out in. There are several limitations in existing approaches. Garg used 3D images in his method to recognize the hand gesture, but this process is complex and also not time efficient. The Processing time is one of the very critical factors in real time applications. Aznaveh presented an RGB vector-based method for skin detection in images. Yang analyses the hand contour to select fingertip candidates, then finds peaks in their spatial distribution and checks local variance to locate fingertips. This method is not invariant to the orientation of the hand. There are other methods, which are using directionally Variant templates to detect fingertips. Few other methods are dependent on specialized instruments and setup like the use of infrared camera, stereo camera, a fixed background or use of markers on hand. This paper describes a novel method of motion patterns recognition generated by the hand without using any kind of sensor or marker.

The fingertips detection should be near to real time if it is going to process video. Generally, image-based models work on pixel by pixel and do hand Segmentation and work only on region of interest. However, most hand segmentation methods can’t do a clearly hand segmentation under some conditions like fast hand motion, cluttered background, poor light condition. If the hand segmentation is not valid, then detection of fingertips can be questionable. Researchers used infrared camera to get a reliable segmentation. Few researchers in their work limit the degree of the background clutter, finger motion speed or light conditions to get a reliable segmentation. Raheja and few others also used 3D mapping using specialized device like KINECT for hand segmentation. Some of fingertip detection methods can’t localize accurately multidirectional fingertips. Researchers assumed that the hand is always pointing upward to get precise localization.

2.1.2 Existing System and Solutions

We began by examining the recent papers from the University of Canterbury, where work, similar to that proposed in this paper, has been developed in the past. For example, Lang produced a webcam-based piano application – however the focus of this was on entertainment, not education, and there was no attempt made to detect the keyboard or keys of a real piano.

Deaker proposed a piano tutoring system which would test the user's knowledge by asking them to play particular notes or chords on a paper keyboard with a printed design [4]. Deaker's proposal is similar to ours in that it requires the detection of the printed keyboard's individual keys in order to determine if the user has followed instructions correctly. However, Deaker's design has several limitations, perhaps the most significant of which being that the identification of keys is not entirely achieved by computer vision alone. A particular number of keys are required to be visible to the camera during calibration (14 white keys, starting from a C key and ending on the B key just under two octaves above) and the software assumes that this requirement has been met.

Another computer vision based piano tutor was proposed by Dawe [6]. In this case, the system is intended for relaying a student's choice of fingering to their teacher during a remote piano lesson. Similarly, to Deaker's proposal, Dawe's method requires a calibration stage where the user is required to select a region of interest – in order to narrow the video frames down to the area occupied by the instrument's keyboard. This brings with it the same limitations imposed by Deaker's method; that is the camera must remain stationary at all times, and the keyboard must not be obscured during the calibration process.

To find the key locations during the calibration phase, Dawe used horizontal and vertical edge detection to determine the boundaries of the keyboard and its individual keys. Unfortunately, a major limitation of Dawe's method was that a translation between these detected edges and a description of the location of each key of the keyboard was unable to be implemented.

After examining these proposed methods, it was clear that a satisfactory method of keyboard and key detection had not been found. The following works were more successful in their attempts to detect individual piano keys. The work by Barakonyi and Schmaltzier demonstrates a variety of 3D augmented reality (AR) applications, including the Augmented Piano Tutor application [2]. The application displays an AR overlay on the user's electronic keyboard, indicating which keys are to be played.

Unfortunately, the paper by Barak Onyi and Schmaltzier does not describe their implementation details – making it difficult to learn any lessons from their AR solution. However, some of their high-level implementation is explained, and even from this it is possible to suggest some improvements – albeit minor ones. The Augmented Piano Tutor application makes use of AR Toolkit for 3D registration. The use of AR Toolkit means that a fiducial marker is required as a reference point for the application. It would be most convenient for the user if they did not have to correctly place a marker or do any more than set up their webcam and laptop, to use a tutoring program. An ideal system would allow the user to simply run an application, point their webcam at a keyboard, and begin playing.

The method proposed by Gorodnichy and Yogeswaran attempts to solve the same problem as identified by Dawe; that is, Gorodnichy and Yogeswaran present a solution to allow current remote piano teaching systems to communicate fingering information [12]. There is one major limitation of Gorodnichy and Yogeswaran's method; the camera used is not an off-the-shelf webcam, but instead a more expensive video camera with high quality output and zoom functionality is required.

The final keyboard detection method we shall discuss is from a paper by Huang, Zhou, Yu, Wang and Du, where a marker less AR piano system is proposed. The 3D AR result is very similar to that achieved by Barak Onyi and Schmaltzier, although in this case a fiducial marker is not required [8]. To identify the keys of the keyboard, a similar method to that of Gorodnichy and Yogeswaran is employed. After identifying the piano keys, this solution uses OpenGL to augment the video stream with 3D graphics.

This was a paper that seemed to accomplish almost all of what we were setting out to achieve. Consequently, the work by Huang, Zhou, Yu, Wang and Du is regarded as a target result for our paper, with our application intended to match the results produced by the marker less AR piano system as best as possible. More specifically, the system by Huang et al. managed the keyboard identification and rendering of the video feed, with overlays, at a frame rate of approximately 15 frames per second. We hoped to match or exceed this performance with our solution.

As a final note on the system developed by Huang et al., there is a small way in which the design could arguably be improved upon: The marker less AR piano system requires a one-time calibration for each camera and/or piano used, and for the 3D registration to be possible, the dimensions of the keyboard need to be known by the software. This requires that the user manually measure their piano keyboard, then input the information into the AR program. At the expense of 3D augmentations, this potential inconvenience can be removed.

2.1.3 Research Findings for Existing Literature

S. No.	**Roll Number**	**Name**	**Paper Title**	**Tools/ Technology**	**Findings**	**Citation**
1	101683035	Akshay Sharma	Marker-less piano fingering recognition using sequential depth images	Python and OpenCV	Found how camera is recognizing keys to play tone.	Adam Goodwin, Richard Green, "Key detection for a virtual piano teacher", Image and Vision Computing New Zealand (IVCNZ) 28th International Conference of, pp. 282-287, 2013.
2	101683035	Akshay Sharma	Major Piano Triad Chords	Python and OpenCV	Found how real time sensing helps to play the associated piano tone.	Mohammad Akbari, Howard Cheng, "Real-Time Piano Music Transcription Based on Computer Vision", Multimedia IEEE Transactions on, vol. 17,

						no. 12, pp. 2113-2121, 2015.
3	101683035	Akshay Sharma	An intelligent virtual piano tutor	MATLAB and OpenCV	Found how computer vision helps to recognize fingers.	Chutisant Kerdvibulvech, *Augmented Reality, Virtual Reality, and Computer Graphics*, vol. 10325, pp. 251, 2017.
4	101683035	Akshay Sharma	A Computer Vision Method of Piano Tutoring	Python	Found how Raspberry pi works for real time motion of fingers with the help of the camera.	D. Schwarz, S. Britton, R. Cahen, T. Goepfer. Musical Applications of Real-Time Corpus-Based Concatenative Synthesis. International Computer Music Conference (ICMC) (Copenhagen, Denmark), pp. 23, 2007.
5	101683035	Akshay Sharma	Virtual Piano Design via Single-View Video Based on Multifinger Actions Recognition	MATLAB	Found what should be the positioning of the keys and how to optimize the speed of sensing.	A. Freed and R. Avizienis, “A new music keyboard featuring continuous key-position sensing and high-speed communication options,” in Proceedings of the International Computer Music Conference, Berlin, Germany, 2000.
6	101683035	Akshay Sharma	Computer Vision Piano Tutor	Python, Adafruit	Found how a real time system actually works for playing	Mohammad Akbari, Jie Liang, Howard Cheng, “A real-time system for online learning-based visual transcription of piano music”, Multimedia Tools

					piano music.	and Applications, vol. 77, pp. 25513, 2018.
7	101503008	Abhishek Sharma	A Markerless Augmented Reality Based Piano Teaching System	MATLAB	Found how we can detect multifinger gesture sensing to play associated piano tone.	Multidimensional gesture sensing at the piano keyboard," in Proceedings of the 29th ACM Conference on Human Factors in Computing Systems (CHI), Vancouver, Canada, pp. 72, 2011
8	101503008	Abhishek Sharma	The Size of the Piano Keyboard	Python	Found how keyboard piano works.	L. Haken, E. Tellman, and P. Wolfe, "An indiscrete music keyboard," Computer Music Journal, vol. 22, no. 1, pp. 30–48, 1998.
9	101503008	Abhishek Sharma	PianoTouch: A wearable haptic piano instruction system for passive learning of piano skills	Python, Adafruit	Found what are the problems we have to face in virtual piano like we need good lightning conditions.	Boga Vishal, K Deepak Lawrence, "Paper piano — Shadow analysis-based touch interaction", Man and Machine Interfacing (MAMI) 2nd International Conference on, pp. 1-6, 2017.
10	101503008	Abhishek Sharma	Computer vision method for pianist's	Python, OpenCV	Found how camera will detect keys frame by	Adam Goodwin, Richard Green, "Key detection for a virtual piano teacher", Image and Vision

			fingers information retrieval		frame using image processing	Computing New Zealand (IVCNZ) 28th International Conference of, pp. 282-287, 2013
11	101503008	Abhishek Sharma	Musical Desktop: A Webcam Piano	Structured Query Language	Found how raspberry pi fetches the correct tone from the database.	C. von Hardenberg, F. Berard, “Bare-hand human-computer interaction”, Proceedings of the 2001 workshop on Perceptive user interfaces, pp. 1-8, 2001.
12	101503008	Abhishek Sharma	Detection and tracking of pianist hands and fingers	MATLAB	Found how to take accurate measurement of moving fingers in real time.	T. Yang, S. Z. Li, Q. Pan, J. Li, “Real-time and accurate segmentation of moving objects in dynamic scene”, Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks, pp. 136-143, 2004.

Table 2.1: Research Findings for Existing Literature

2.1.4 Problem That Has Been Identified

Moving piano from one home to another without the utmost care can easily cause unforeseen problems. Hence Pianos are large instruments that cannot be carried everywhere. Even electric pianos need to be carried in a large bag and are prone to damage in travelling.

Today, the piano buyer needs to expect to pay a minimum of $1,200-1,500 for a piano that will hold tune and plays reasonably well and doesn't look horrible in your living room. Most used pianos that are for sale are on the market because the family has lost interest in playing the piano.

So here we propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it. The solution mainly focuses on the cost and budget of the Music Enthusiastic [14]. Virtual Piano will reduce the problem of carrying large sized pianos.

2.1.5 Survey of Tools and Technology Used

Video is the sequence of the image frames at a fixed rate. First of all, Images would be captured with a simple camera in 2D continuously and would be processed one by one. An HSV colour space-based skin filter would be applied on the images for hand segmentation. An intensity-based histogram would be constructed for the wrist end detection and image would be cropped so that resultant image would have only hand pixels. The fingertips would be detected in the cropped hand image and would be marked differently.

Skin Filter

An HSV colour space-based skin filter would be used on the current image frame for hand segmentation. The skin filter would be used to create a binary image with black background. This binary image would be smoothened using the averaging filter. There can be false positives, to remove these errors the biggest BLOB (Binary Linked Object) is considered as the hand and rest are background. The biggest BLOB represents hand coordinates "1" and "0" to the background [4]. The filtered-out hand image, after removing all errors. The only limitation of this filter is that the BLOB for hand should be the biggest one.

Wrist End Detection

To crop the image, we need to find out the direction of hand and wrist end. First of all, an intensity histogram of the binary silhouette would be formed. Histograms functions are: Here imb represents the binary silhouette and m, n represents the row and columns of the matrix imb.

$$H_x = \sum_{y=1}^{n} imb(x,y)$$

$$H_y = \sum_{x=1}^{m} imb(x,y)$$

To find the direction of hand, after a 4-way scan of image, we choose the maximum value of on pixels coming out of all scans ("1" in the binary silhouette). it was noted that maximum value of "ON" pixels represents wrist end and opposite end of this scan end would represent the finger end. Similarly, the green bar corresponds to right scan, red bar corresponds to down scan, and pink bar corresponds to up scan of „on" pixels in the binary silhouette. Now, it is clear that red bar has higher magnitude than other bars. So, we can infer that the wrist end is in downward direction of the frame and consequently the direction of fingers is in the upward direction [8]. Here the direction from wrist to finger is known.

Hand Cropping

Hand cropping minimizes the number of pixels to be considered for processing which leads to minimization of computation time. In the histogram generated, it was observed that at the point where wrist ends, a steeping inclination of the magnitude of the histogram starts, whose slope, m can be defined as:

$$m = \frac{y2-y1}{x2-x1}$$

The scan would start from the end where skin pixel was found and move till an inclination would be found. The points correspond to the first skin pixel scanning from other three sides were found, crop the image at that points. The equations for cropping the image are:

$$\text{imcrop} = \begin{Bmatrix} origin_{image}, for\ Xmin < X < Xmax \\ 0 \qquad\qquad\qquad\qquad elsewhere \end{Bmatrix}$$

Where imcrop represents the cropped image, Xmin, Ymin, Xmax, Ymax represent the boundary of the hand in the image. Some results with processing steps for hand cropping. The arrows showed in the main frames indicate the direction of scan which was found from wrist end detection step [7]. In all the histograms it is clear that at the wrist point, a steeping inclination starts in the scanning direction.

Fingertip Detection

At this point we have one smaller image which conation only skin pixels (hand gesture shape). We will figure out fingertips in the cropped hand image. Start scan the cropped binary image from wrist to fingers end and calculate the number of pixels for each row or column based on the hand direction in up-down or left-right. Assign the intensity values for each pixel from 1 to 255 in increased manner from wrist to finger end in equal distribution. So, each "on" pixel on the edges of the fingers would be assigned a high intensity value of 255. Hence all the fingertips contain pixel value 255. The fingertip detection can be represented mathematically as;

$$\text{pixel}_{\text{count}}(\text{y}) = \sum_{X=Xmin}^{Xmax} imb(x, y)$$

$$\text{modified\_image (x, y)} = \text{round}(\text{x}*255/\text{pixel}_{\text{count}}(\text{y}))$$

$$\text{Finger}_{\text{edge}}(\text{x,y}) = \begin{cases} 1 \; if \; modified_{image}(x, y) = 255 \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases}$$

The line having high intensity pixel, is first indexed and check whether differentiated value lie inside a threshold, if it is then it represents a fingertip. The threshold value changes toward the direction of hand. That threshold can be set after the detection of the direction of hand to the finger which we already know. where detected pixels are marked with different color. any direction, only palm should face the camera [11].

The fingertips would be detected irrespective of user orientation. The movement of user's finger will control the robot hand and its working, by moving hand in front of camera without wearing any gloves or markers.

2.2 Standards

Industry 4.0: Industry 4.0 is a name given to the current trend of automation and data exchange in manufacturing technologies. It includes cyber-physical systems, the Internet of things, cloud computing and cognitive computing. Industry 4.0 is commonly referred as the fourth industrial revolution. Industry 4.0 fosters what has been called a "smart factory". Within modular structured smart factories, cyber-physical systems monitor physical processes, create a virtual copy of the physical world and make decentralized decisions. Over the Internet of Things, cyber-physical systems communicate and cooperate with each other and with humans in real-time both internally and across organizational services offered and used by participants of the value chain.

2.3 Software Requirements Specification

2.3.1 Introduction

2.3.1.1 Purpose

We propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried anywhere and does not have any electronic components in it. Large Pianos are prone to damage in travelling and these Virtual Pianos will take care of this issue. Virtual Pianos will be much cheaper than traditional pianos. It will help young children to get a feel for music and will act as a stepping stone for some of the world's greatest artists.

2.3.1.2 Intended Audience and Reading Suggestion

1. Developers who can review project's capabilities and more easily understand where their efforts should be targeted to improve or add more features to it (design and code the application – it sets the guidelines for future development).

2. Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes more methodically organized.
3. End users of this application who wish to read about what this project can do.

2.3.1.3 Project Scope

Moving piano from one home to another without the utmost care can easily cause unforeseen problems. Hence Pianos are large instruments that cannot be carried everywhere. Even electric pianos need to be carried in a large bag and are prone to damage in travelling. Today, the piano buyer needs to expect to pay a minimum of $ 1,200-1,500 for a piano that will hold tune and plays reasonably well and doesn't look horrible in your living room. Most used pianos that are for sale are on the market because the family has lost interest in playing the piano.

So here we propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it. The solution mainly focuses on the cost and budget of the Music Enthusiastic. Virtual Piano will reduce the problem of carrying large sized pianos. The Scope of the project mainly focuses on the following things: -

1. Large Pianos are prone to damage in travelling and these Virtual Pianos will take care of this issue.
2. Virtual Pianos will be much cheaper than traditional pianos.
3. It will help young children to get a feel for music and will act as a stepping stone for some of the world's greatest artists.

2.3.2 Overall Description

2.3.2.1 Product Perspective

Virtual reality is all about bringing your imagination into existence. our focus is to create virtual- reality-based-products, and bring the concept of "conceptualize and virtualize". "virtual piano" expanding to " virtual piano " is one of our products that limelight this concept by allowing users to play piano by just touching on a 2d piano image. cooked on the base of computer vision, with extracts of python and essence of image processing makes it a full-fledged techno-pie that brings virtual reality and shows its advantages. Virtual piano, sees the image of piano in real world using a webcam, implemented using python computer vision and when a user touches on image the roi (region of interest) is calculated using image processing that which key is pressed by finger (based on a unique centroid based pattern matching algorithm) and accordingly the corresponding frequency of that note is determined and corresponding sound is played. This explains that innovation and technology is what matters for virtual piano. Thus, we are allowing users to eliminate the use of hardware and use virtual reality to play piano anywhere, anytime by just touching its image. our basic strategy is to eliminate the use of hardware by infusing life in images using virtual reality. Hence virtual piano will bring a total revolution in the music industry through the use of technology. It can be used as a midi keyboard also and thus finds its scope in the music industry.

Necessity

The need for the product is recognized. The decision to provide the product is made. The scope of the product is determined. The requirements for the product are gathered.

Invention

The product is designed from the requirements. The product is created from the design. The product is documented and undergoes testing

Release

The product is packaged for outside release. The product goes through further steps of external testing. The product may be handed off to an operational group.

Support

An operational group may support the product. A user service group may support the product. Development may support operations. Development will support the code.

DE support

The lack of need for the product is recognized. The product may be phased out entirely, or support may transition to a different product.

2.3.2.2 Product Features

Music has always been an important part of worship for Latter-day Saints. It inspires and strengthens, brings beauty and unity, and is a unique way to express feelings about the gospel. Many Church members want to learn how to read music, conduct hymns, and play a keyboard instrument. The purpose of the Basic Music Course is to help you develop these skills. As you do, you will enrich your life and be able to serve in new ways. The Basic Music Course has two parts: The Conducting Course and the Virtual Piano Course. You do not need previous musical training to begin these courses. As you progress through them, you will learn music skills in a carefully planned order. You should begin with the Conducting Course. After completing it, you will know the basics of rhythm and note reading; you will also know how to use the Church hymnbook and conduct most hymns. After completing the Virtual Piano Course, you will know how to read music and play some simple hymns on an instrument. The Basic Music Course can be used in branches, wards, stakes, and homes to teach all interested members and nonmembers. No fees beyond the cost of materials should be charged.

We propose a method for identifying a piano keyboard present in the video footage of a standard webcam with the goal of teaching chords, scales and suggested finger positions to a beginner pianist. Our keyboard identification method makes use of binary thresholding, Sobel operators and Hough transforms, as well as proposed algorithms specific to this application, to first find an area resembling a piano keyboard before narrowing the search to detect individual keys. Through the use of our method the keys of a piano keyboard were successfully identified from webcam video footage, with a tolerance to camera movement and occluded keys demonstrated. This result allowed the augmented reality style highlighting of individual keys, and the display of suggested fingering, for various chords and scales.

2.3.3 External Interface Requirements

2.3.3.1 Hardware Interfaces

LCD Display

Frequently, an 8051 program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an 8051 is an LCD display. Some of the most common LCDs connected to the 8051 are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Fortunately, a very popular standard exists which allows us to communicate with the vast majority of LCDs regardless of their manufacturer. The standard is referred to as HD44780U, which refers to the controller chip which receives data from an external source (in this case, the 8051) and communicates directly with the LCD.

Figure 2.1: LCD Display

Raspberry Pi

The Raspberry Pi 3 is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

The Raspberry Pi 3 is the third generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016. Compared to the Raspberry Pi 2 it has:

1. A 1.2GHz 64-bit quad-core ARMv8 CPU
2. 802.11n Wireless LAN
3. Bluetooth 4.1
4. Bluetooth Low Energy (BLE)

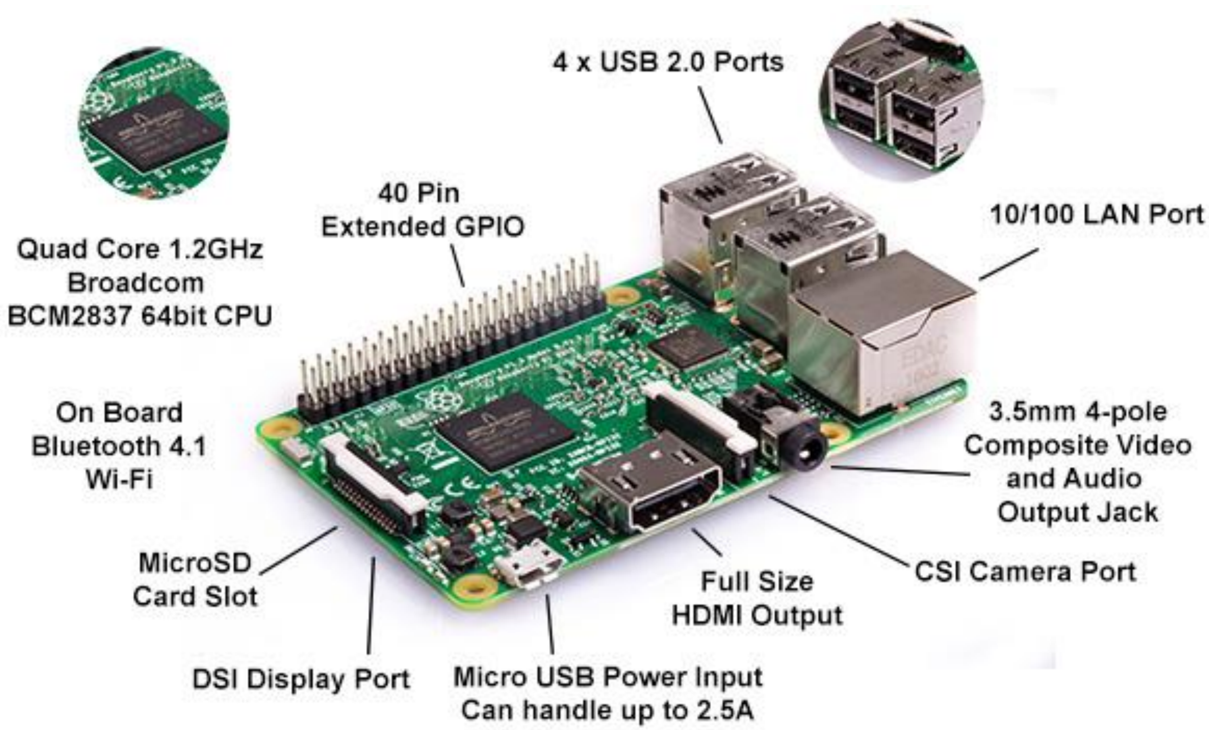

Figure 2.2: Raspberry Pi

Plastic Sheet

We have used a semi-transparent plastic sheet for making this project. This plastic sheet is basically displaying us the seven keys of our piano i.e. C, D, E, F, G, A and B. The plastic sheet is basically serving as a replacement of actual piano. This plastic sheet is just used to give people a feel of real piano as there is no need for using this plastic sheet for constructing our Virtual Piano. We have basically used Image Processing to divide our plastic sheet into different segments so whether we use the plastic sheet or not is just a matter of our choice.

This plastic sheet is made up of rigid and semi-transparent material of size 8 inch * 10 inch and thickness 0.5 cm. It is basically used to support the wooden stick on which camera is mount.

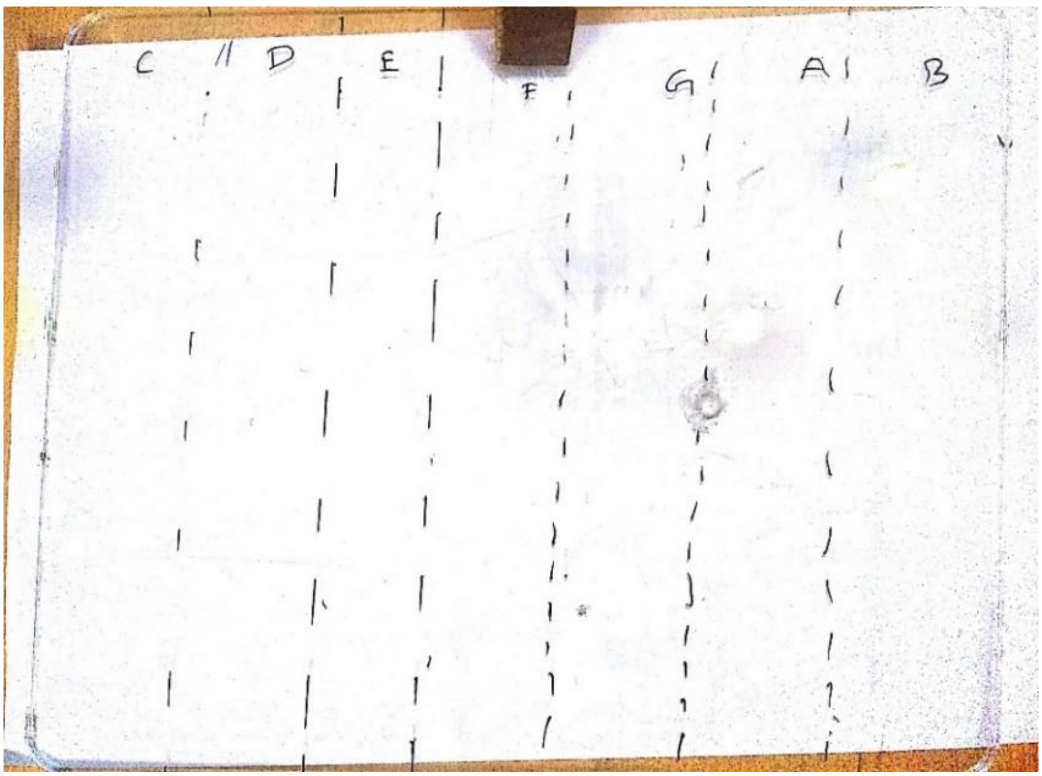

Figure 2.3: Plastic Sheet

Speaker

A simple speaker has been used for playing the tones which have been simulated for each section by detecting human fingers through the plastic sheet.

Figure 2.4: Speaker

2.3.3.2 Software Interfaces

IDLE is Python’s Integrated Development and Learning Environment.

IDLE has the following features:

1. Coded in 100% pure Python, using the tkinter GUI toolkit.
2. Cross-platform: works mostly the same on Windows, Unix, and Mac OS X.
3. Python shell window (interactive interpreter) with colorizing of code input, output, and error messages.
4. Multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features.

5. Search within any window, replace within editor windows, and search through multiple files (grep).
6. Debugger with persistent breakpoints, stepping, and viewing of global and local namespaces.
7. Configuration, browsers, and other dialogs.

Python is a wonderful and powerful programming language that's easy to use (easy to read and write) and with Raspberry Pi lets you connect your project to the real world. Python syntax is very clean, with an emphasis on readability and uses standard English keywords. Start by opening IDLE from the desktop.

The easiest introduction to Python is through IDLE, a Python development environment. Open IDLE from the Desktop or applications menu:

IDLE gives you a REPL (Read-Evaluate-Print-Loop) which is a prompt you can enter Python commands in to. As it's a REPL you even get the output of commands printed to the screen without using print.

2.3.4 Other Non-functional Requirements

2.3.4.1 Performance Requirements

The only way in which systems will meet their performance targets is for them to be specified clearly and unambiguously. It is a simple fact that if performance is not a stated criterion of the system requirements then the system designers will generally not consider performance issues. While loose or incorrectly defined performance specifications can lead to disputes between clients and suppliers. In many cases performance requirements are never ridged as system that does not fully meet its defined performance requirements may still be released as other consideration such as time to market.

In order to assess the performance of a system the following must be clearly specified:

1. Response Time
2. Workload
3. Scalability
4. Platform

Response Time

In some cases, the system response times are clearly identified as part of a business case for example a criminal's fingerprint need to be identified while the criminal is still in custody (less than an hour). In some cases, the response time will be dictated by legal requirements although this is rare.

For general applications asking users to what is an acceptable response time is like asking people how much salary they require!! The whole process is simply a process of negotiation.

The general advice on response time from Jakon Nielsen book on Usability is:

0.1 second is about the limit for having the user feel that the system is reacting instantaneously, meaning that no special feedback is necessary except to display the result. 1.0 second is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data. 10 seconds is about the limit for keeping the user's attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

For systems that have to support significant numbers of users the cost of response times delays can actually be measured in monetary terms and therefore can form part of trade-off studies between different architectures providing different levels of performance.

Whatever is chosen must be measurable in the real system. Care must be taken to ensure that the performance measurement is unambiguous, concise and completely defined. Response time specification should include the following information: Measurement Points: The points at which you want the response time measured needs to be included. For example, are you interested in response time at the data center or from a branch network?

What is included: Define what is included in the measurement. For example, does the measure for a web page include the browser render time or just the delivery time to the browser?

What is excluded: Ensure calls to 3rd parties beyond the control of the system developer are defined. This is particularly important when defining contractual response time requirements, as suppliers can only be responsible for areas they can influence.

Statistic Type: The statistic type needs to be defined. For example, 95% of all response time should be less than 8.5 second.

Measurement Period: Define the period over which to measure the response time. This is particularly important where workload varies over the day.

Platform: Should it not be possible to test on the real production hardware any allowance for testing on test hardware needs to be stated.

Error Rate: Define the acceptable error rate allowed during the measurement of the response times. Some systems may produce errors under high workloads and therefore the acceptable error rate need to be defined.

Finally, once you have defined the response times have them reviewed by somebody else to check if the definition is indeed clear and unambiguous. Remember also to state the workload at which the response time are to be met.

Workload

Again, the business case or existing process should be the start of the workload definition. However, it is not enough to state that “the system should be capable of supporting 80,000 customers” or “the system should be able to support 4 pages/sec”. These statements are often good metrics at a high-level management level but do not define the work that the system must support. This is particularly important as the mix of transaction affects the performance. For example, a DB system may easily handle 10,000 read transaction per hour but only 3,000 update transactions per hour.

The most likely transactions to specify are the user-initiated transactions but care must be taken to consider all the users of the system and the batch processes.

For example, a system may have external customers, internal staff providing data entry and batch processes such as backups. If the backup is not completed overnight, then it may seriously disrupt the performance experienced by the users the next day.

The workload is often described as the scenarios that the users are likely to execute. The table below is an example that shows for a user scenario how many requests per day, what pages a user executes and the think time between pages. You will notice that a percentage is included before the exit page, this is to specify that only a percentage of users will view that page.

When completing a workload specification, a check must be made to ensure that all relevant functions have been covered. This includes not just the obvious user workloads but special cases such as management requests, backups and error scenarios/handling. Once all loads have been considered, infrequent or inappropriate workloads can be eliminated. Inappropriate workloads might include error scenarios where it is understood that errors will be very infrequent although it should be appreciated that an appreciable workload can result from the need to provide adequate error and/or security logging.

The workload should be specified up to the date that you wish current hardware to support without upgrade.

The performance of the system is dependent on how the load is delivered to the system. For example, it is easier to achieve faster response times for a system that receive a regular arrival of work throughout the day compared to one that receive burst of traffic. Therefore, it is important that the workload profile is defined

The arrival rate into the system is rarely going to be constant throughout the day or the week. The figure below shows the arrival rate for requests to a website.

As can be seen in the figure the workload peaks around lunchtime and late evening but activity is very quiet during the night. The above example shows how workload varies over a 24-hour period but it may be more important to show how workload varies over a month for example batch or over an hour for example end of day share trading.

Many organizations' workload may suffer a rare or unexpected increase in the workload of the system and this is often referred to as a "peak" workload. The decision facing system developers is whether to design a system to cope with a peak workload. The answer to this will depend on the consequence of failing. For example, certain organizations design for a peak. The NY stock exchange in 1997 could process 3 billion transactions in a day while the daily average is less than 1.2 billion (Times, August 14 1997). This is because of the need to execute trades the same day and to maintain an accurate record of the state of the market.

When defining the workload for a new venture and little or no existing workload data exists and the system is to be developed by a supplier the temptation is to specify a high peak workload. This is a valid but often an expensive approach which may still not protect you from high demand as it can still be higher than you specify! An alternative solution is to specify the workload as your business model and analysis predicts it will be. Next, have a detailed specification of the scalability of the system to ensure that the supplier develops a system that is scalable quickly. Specify requirements for detecting and processing overload to ensure flood control mechanisms are in place to avoid the system crashing under intensive loads. Test that the system is scalable by renting in additional equipment or using vendors test environments. Finally ensure the operational staff have a procedure for increasing the scalability of the site and contracts are in place for additional hardware/bandwidth etc.

Scalability

In one respect scalability is simply specified as the increase in the system's workload that the system should be able to process. The scalability required is often driven by the lifespan and the maturity of the system. For example, a new (and hence immature) system could suffer an unexpected growth in popularity and suffer from a significant increase in workload as it becomes popular with new users. More mature systems which represent improvements on older systems are likely to have more accurately defined workloads and thus be less likely to suffer in this respect, remember to specify that the response time requirements should still be meet as the workload scales A problem with scalability specification is that it may not be economically viable to test the scalability, as it often requires additional hardware. Therefore, an alternative is either to rent in the additional hardware for the purpose of the tests or to use an extrapolation technique such as a simulation model.

Platform

A platform is defined as the underlying hardware and software (operating system and software utilities) which will house the system. It is not always the case that the designer will be given a "green field" choice of what platform on which to house the system. In some cases, the customer may dictate this choice or there may be internal reasons (product strategy perhaps) that will constrain the designer's freedom. It may also be the case that the system will require various generic products to be used in which case the performance of these must also be specified.

Consideration must be given as to whether the hardware will be used exclusively for the system or whether it must be shared with some other process. A new system might have to compete with an old system for the processor and its resources. The processor and resource requirements of the old system must be specified so the amount of free space and performance can be assessed.

When the customer specifies the platform for the system it is not only important that the project engineer knows what has been specified but that they also understand its capability in terms of performance.

When part of the platform consists of external resources such as connections to external databases or banking systems then the response times of these external resources must also be specified.

2.3.4.2 Safety Requirements

1. Do not touch Raspberry Pi or any component on the PCB board with wet hands.
2. The setup should be prevented from any water spill.
3. There are limited power requirements. The supply should not be more than 5A.
4. The system should be dismantled properly after using the piano.
5. All the components should be kept at one place as there are many small components like micro sd card and if any one of them is misplaced, it will create a huge problem.
6. The system as a whole should be handled with care.

2.3.4.3 Security Requirements

The analysis is the most essential activity to obtain the understanding between the development team and the business team. It maps the information to develop systems/applications in accordance to business and user needs, provided by stakeholders as high-level statements on features and functionalities.

In order to cover security aspects, it is necessary to bring together business, development and security teams to understand the key sensitivities and business consequences caused by risk of security flaws. Since SDLC is a feed forward process and errors introduced in this phase will be spread in the next development phases, it is important to analyze security risks at very early stages.

Currently security requirements are categorized as nonfunctional requirements, which are usually defined as attributes of the software, skipping to be mapped to proper functional requirements and therefore, may not built into the software neither tested appropriately. However, what is the security impact if the security requirements are not captured or defined? Probably the result application may not be assessed for success or failure prior to implementation. Map the nonfunctional to functional requirements, the security requirements become a part of the overall requirements analysis process and, in this case, if conflicts are inevitable than they need to be identified and treated properly.

To show how this discussion has high security impact for application, the IATAC published the report Software Security Assurance (Goertzel, et al., 2007) where the authors met the main security-specific issues involved in requirements engineering, among them:

1. The people involved are not likely to know or care about non-functional requirements. Stakeholders tend to take for granted non-functional security needs.
2. Traditional techniques and guidelines tend to be more focused on functional requirements.
3. Security controls are perceived to limit functionality or interfere with usability.
4. Stakeholders must understand the threats facing a system in order to build defenses against them.
5. The users who help define the system are not typically the abusers from whom the system must be protected.
6. It is more difficult to specify what a system should not do than what it should do.

2.4 Cost Analysis

Raspberry PI:	₹ 3,649.00	**Digital Multimeter:**	₹ 250.00
PCB Board:	₹ 185.00	**LCD Display**:	₹ 1410.00
SD Card:	₹ 599.00	**Wire stripper & cutter**:	₹ 58.00
Speaker:	₹ 200.00	**Miscellaneous:**	₹ 399.00
Webcam:	₹ 850.00		
Soldering Iron:	₹ 299.00		
Total	₹ 7899.00		

Table 2.2: Cost Analysis

2.5 Risk Analysis

The risk analysis process follows the USACE Headquarters requirements as well as the guidance provided by the Cost Engineering Directory of Expertise for Civil Works (Cost Engineering DX). The risk analysis process reflected within the risk analysis report uses probabilistic cost and schedule risk analysis methods within the framework of the Crystal Ball software. The risk analysis results are intended to serve several functions, one being the establishment of reasonable contingencies reflective of an 80 percent confidence level to successfully accomplish the project work within that established contingency amount. Furthermore, the scope of the report includes the identification and communication of important steps, logic, key assumptions, limitations, and decisions to help ensure that risk analysis results can be appropriately interpreted.

Risk analysis results are also intended to provide project leadership with contingency information for scheduling, budgeting, and project control purposes, as well as provide tools to support decision making and risk management as the project progresses through planning and implementation. To fully recognize its benefits, cost and schedule risk analyses should be considered as an ongoing process conducted concurrent to, and iteratively with, other important project processes such as scope and execution plan development, resource planning, procurement planning, cost estimating, budgeting, and scheduling.

Identifying the risk factors via the PDT are considered a qualitative process that results in establishing a risk register that serves as the document for the further study using the Crystal Ball risk software. Risk factors are events and conditions that may influence or drive uncertainty in project performance. They may be inherent characteristics or conditions of the project or external influences, events, or conditions such as weather or economic conditions. Risk factors may have either favorable or unfavorable impacts on project cost and schedule.

Checklists or historical databases of common risk factors are sometimes used to facilitate risk factor identification. However, key risk factors are often unique to a project and not readily derivable from historical information. Therefore, input from the entire PDT is obtained using creative processes such as brainstorming or other facilitated risk assessment meetings. In practice, a combination of professional judgment from the PDT and empirical data from similar projects is desirable and is considered.

A formal PDT meeting was held at the RAY Building in St. Louis, Missouri, for the purpose of identifying and assessing risk factors. The meeting on March 1, 2010 included the Project Manager, Real Estate, Environmental member, Cost Engineer, and Hydraulic Engineer/Technical Manager Geotechnical Engineer, Cost Engineer, and Hydraulic Engineer/Technical Manager.

The first half of the formal meeting focused on risk factor identification using brainstorming techniques and some facilitated discussions based on risk factors common to projects of similar scope and geographic location. The second half of the formal meeting focused on risk factor assessment and quantification.

Additionally, numerous calls and informal meetings were conducted throughout the risk analysis process on an as-needed basis to further facilitate risk factor identification, market analysis, and risk assessment.

The quantitative impacts of risk factors on project plans are analyzed using a combination of professional judgment, empirical data, and analytical techniques. Risk factor impacts are quantified using probability distributions (density functions), because risk factors are entered into the Crystal Ball software in the form of probability density functions. Similar to the identification and assessment process, risk factor quantification involves multiple project team disciplines and functions. However, the quantification process relies more extensively on collaboration between cost engineering, designers, and risk analysis team members with lesser inputs from other functions and disciplines.

The following is an example of the PDT quantifying risk factor impacts by using an iterative, consensus-building approach to estimate the elements of each risk factor:

1. Maximum possible value for the risk factor.
2. Minimum possible value for the risk factor.
3. Most likely value (the statistical mode), if applicable.
4. Nature of the probability density function used to approximate risk factor uncertainty.
5. Mathematical correlations between risk factors.
6. Affected cost estimate and schedule elements.

Contingency is analyzed using the Crystal Ball software, an add-in to the Microsoft Excel format of the cost estimate and schedule. *Monte Carlo* simulations are performed by applying the risk factors (quantified as probability density functions) to the appropriate estimated cost and schedule elements identified by the PDT. Contingencies are calculated by applying only the moderate and high-level risks identified for each option (i.e., low-level risks are typically not considered, but remain within the risk register to serve historical purposes as well as support follow-on risk studies as the project and risks evolve).

For the cost estimate, the contingency is calculated as the difference between the P80 cost forecast and the base cost estimate. Each option-specific contingency is then allocated on a civil works feature level based on the dollar-weighted relative risk of each feature as quantified by *Monte Carlo* simulation. Standard deviation is used as the feature-specific measure of risk for contingency allocation purposes. This approach results in a relatively larger portion of all the project feature cost contingency being allocated to features with relatively higher estimated cost uncertainty.

For schedule contingency analysis, the option schedule contingency is calculated as the difference between the P80 option duration forecast and the base schedule duration.

These contingencies are then used to calculate the time value of money impact of project delays that are included in the presentation of total cost contingency in section 6. The resulting time value of money, or added risk escalation, is then added into the contingency amount to reflect the USACE standard for presenting the "total project cost" for the fully funded project amount.

Schedule contingency is analyzed only on the basis of each option and not allocated to specific tasks. Based on Cost Engineering DX guidance, only critical path and near critical path tasks are considered to be uncertain for the purposes of contingency analysis.

3.1 Investigative Techniques

S. No.	Investigative Projects Techniques	Investigative Techniques Description	Investigative Projects Examples
1	Descriptive	An investigation was performed in which several real time motion detection algorithms were discussed in detail and after applying different algorithms for our cause, the best one was chosen.	We propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it. Algorithms used are Skin filter, Wrist end detection, Hand cropping, Fingertip Detection.
2	Comparative	Investigations were made here that compared the output of our Virtual Piano with the actual Piano and we were getting the desired output.	Over the past decades, many electronic music controllers have been developed, but few approaches the ubiquity of the piano-style keyboard. Few Algorithms on which keyboard piano works are Raw data frame, Multi-key gestures, Gestural feature extraction.
3	Experimental	Experiment is performed on different test cases such as single finger detection, double finger detection and multi finger detection and the same is verified through LCD Display.	Internet of Things, Augmented Reality and Artificial Intelligence based Projects etc.

Table 3.1: Investigative Techniques

3.2 Proposed Solution

Here we propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried and does not have any electronic components in it. We use a raspberry pi attached to a camera along with the plastic sheet to make a virtual piano. We use image processing to divide the plastic sheet into sections and assign particular tomes to it. We the detect human fingers through the plastic sheet and simulate associate piano tone for each section to play piano tones using a speaker. Thus, we provide a virtual piano which is actually a transparent light weight plastic board that can be carried around roughly.

3.3 Work Breakdown Structure

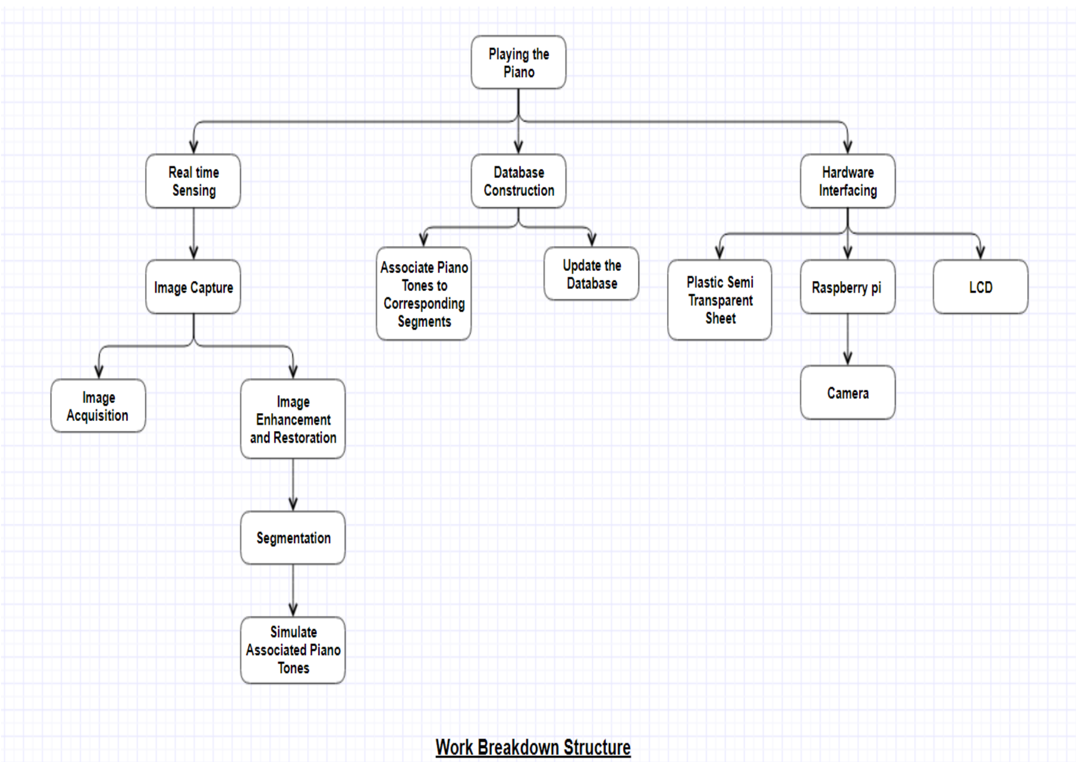

Figure 3.1: Work Breakdown Structure

3.4 Tools and Technologies Used

Video is the sequence of the image frames at a fixed rate. First of all, Images would be captured with a simple camera in 2D continuously and would be processed one by one. An HSV color space-based skin filter would be applied on the images for hand segmentation.

An intensity-based histogram would be constructed for the wrist end detection and image would be cropped so that resultant image would have only hand pixels. The fingertips would be detected in the cropped hand image and would be marked differently.

Skin Filter

An HSV color space-based skin filter would be used on the current image frame for hand segmentation. The skin filter would be used to create a binary image with black background. This binary image would be smoothened using the averaging filter. There can be false positives, to remove these errors the biggest BLOB (Binary Linked Object) is considered as the hand and rest are background. The biggest BLOB represents hand coordinates “1” and “0” to the background [4]. The filtered-out hand image, after removing all errors. The only limitation of this filter is that the BLOB for hand should be the biggest one.

Wrist End Detection

To crop the image, we need to find out the direction of hand and wrist end. First of all, an intensity histogram of the binary silhouette would be formed. Histograms functions are: Here imb represents the binary silhouette and m, n represents the row and columns of the matrix imb.

To find the direction of hand, after a 4-way scan of image, we choose the maximum value of “ON” pixels coming out of all scans (“1” in the binary silhouette). It was noted that maximum value of “ON” pixels represents wrist end and opposite end of this scan end would represent the finger end. Similarly, the green bar corresponds to right scan, red bar corresponds to down scan, and pink bar corresponds to up scan of “ON” pixels in the binary silhouette. Now, it is clear that red bar has higher magnitude than other bars. So, we can infer that the wrist end is in downward direction of the frame and consequently the direction of fingers is in the upward direction [8]. Here the direction from wrist to finger is known.

Hand Cropping

Hand cropping minimizes the number of pixels to be considered for processing which leads to minimization of computation time. In the histogram generated, it was observed that at the point where wrist ends, a steeping inclination of the magnitude of the histogram starts, whose slope, m can be defined.

The scan would start from the end where skin pixel was found and move till an inclination would be found. The points correspond to the first skin pixel scanning from other three sides were found, crop the image at that points. The equations for cropping the image are:

Where imcrop represents the cropped image, Xmin, Ymin, Xmax, Ymax represent the boundary of the hand in the image. Some results with processing steps for hand cropping. The arrows showed in the main frames indicate the direction of scan which was found from wrist end detection step [7]. In all the histograms it is clear that at the wrist point, a steeping inclination starts in the scanning direction.

Fingertip Detection

At this point we have one smaller image which conation only skin pixels (hand gesture shape). We will figure out fingertips in the cropped hand image. Start scan the cropped binary image from wrist to fingers end and calculate the number of pixels for each row or column based on the hand direction in up-down or left-right. Assign the intensity values for each pixel from 1 to 255 in increased manner from wrist to finger end in equal distribution. So, each "ON" pixel on the edges of the fingers would be assigned a high intensity value of 255. Hence all the fingertips contain pixel value 255.

The line having high intensity pixel, is first indexed and check whether differentiated value lie inside a threshold, if it is then it represents a fingertip. The threshold value changes toward the direction of hand. That threshold can be set after the detection of the direction of hand to the finger which we already know. where detected pixels are marked with different color. any direction, only palm should face the camera [11]. The fingertips would be detected irrespective of user orientation. The movement of user's finger will control the robot hand and its working, by moving hand in front of camera without wearing any gloves or markers.

4.1 System Architecture

1. Block Diagram

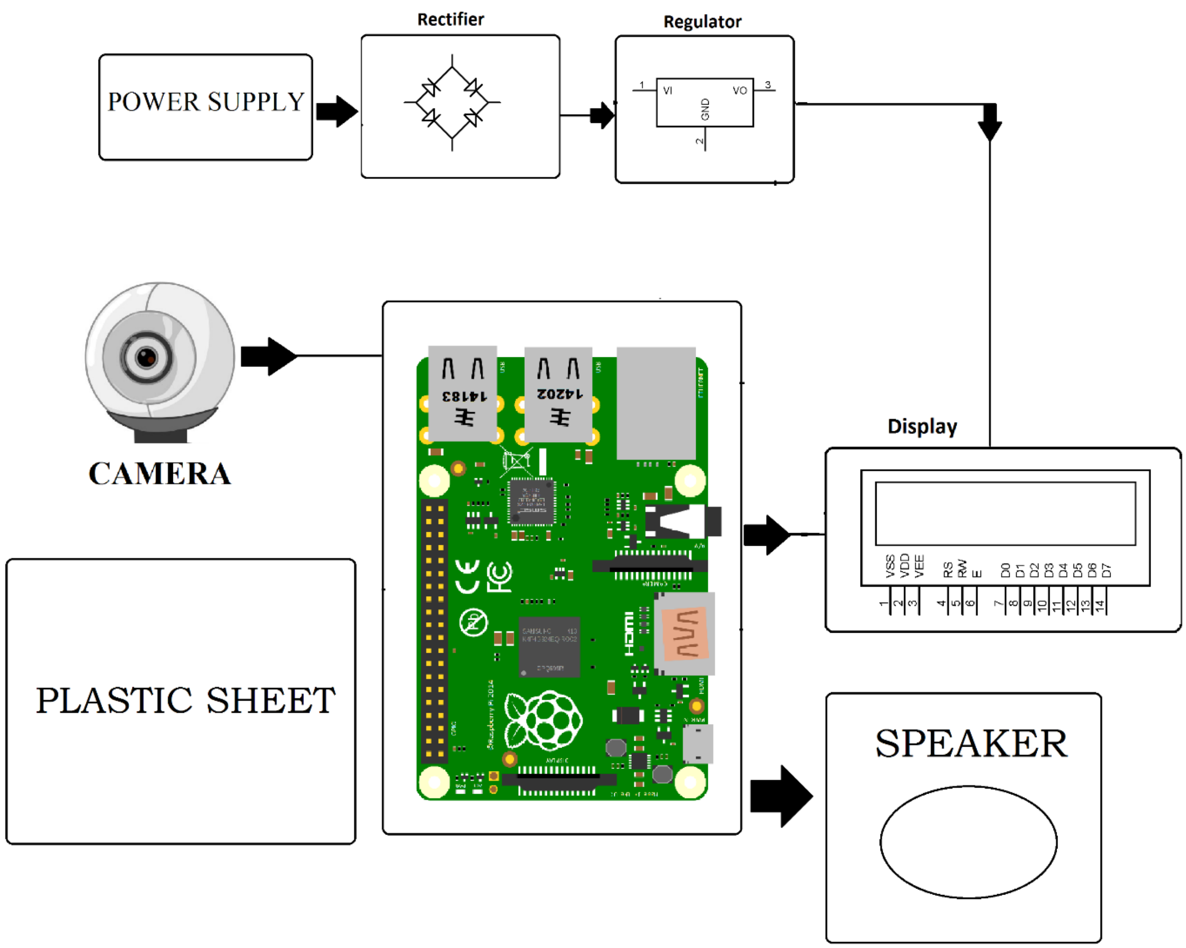

Figure 4.1: Block Diagram

2. Component Diagram

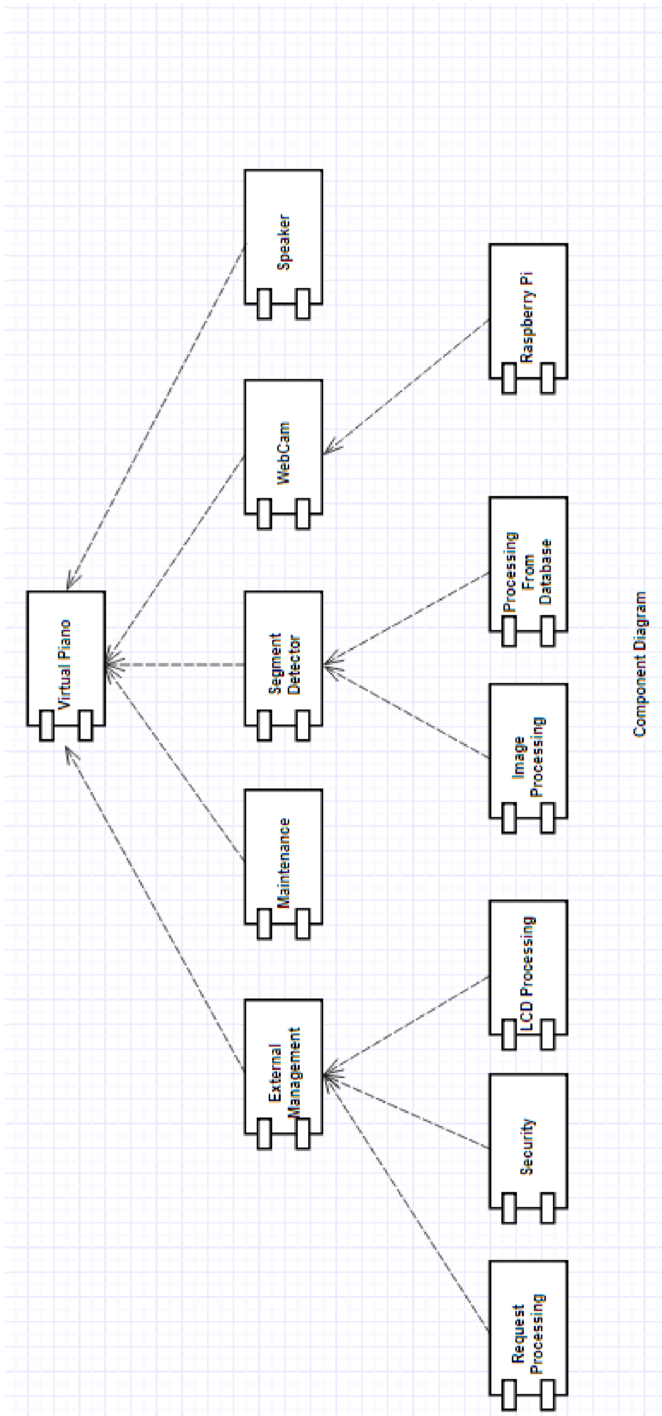

Figure 4.2: Component Diagram

3. Data Centered Architecture

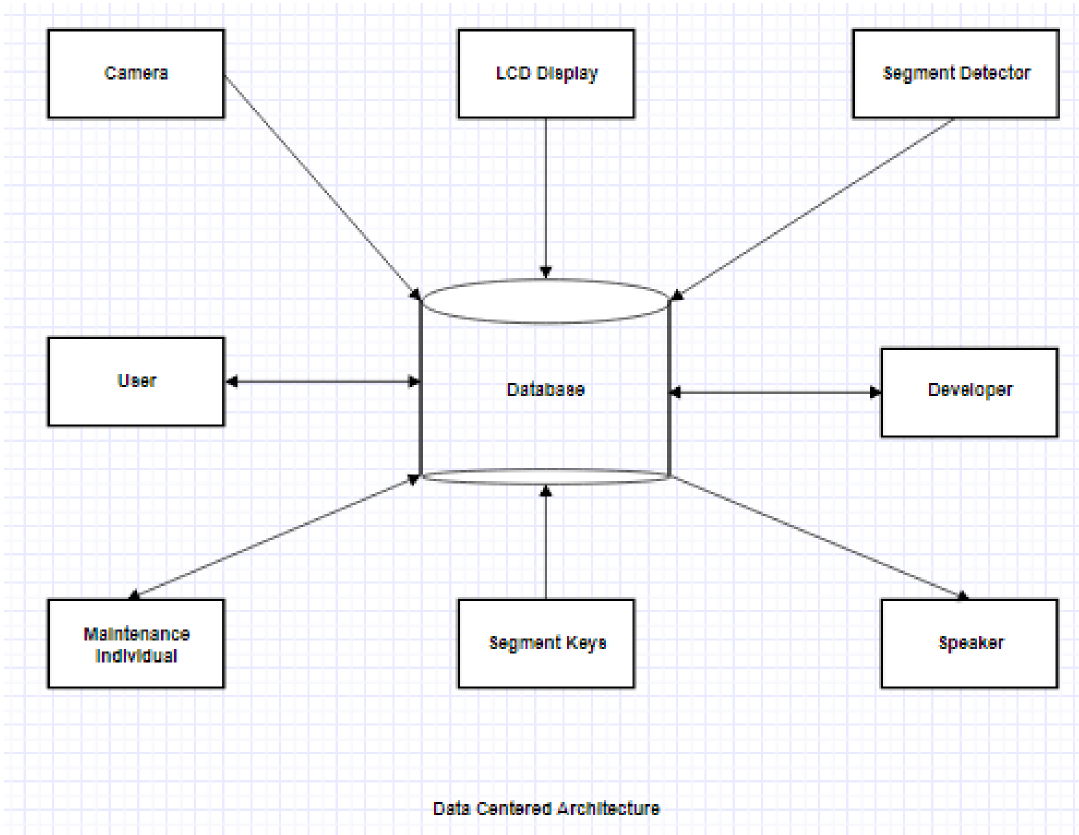

Figure 4.3: Data Centered Architecture

4. Data Flow Architecture

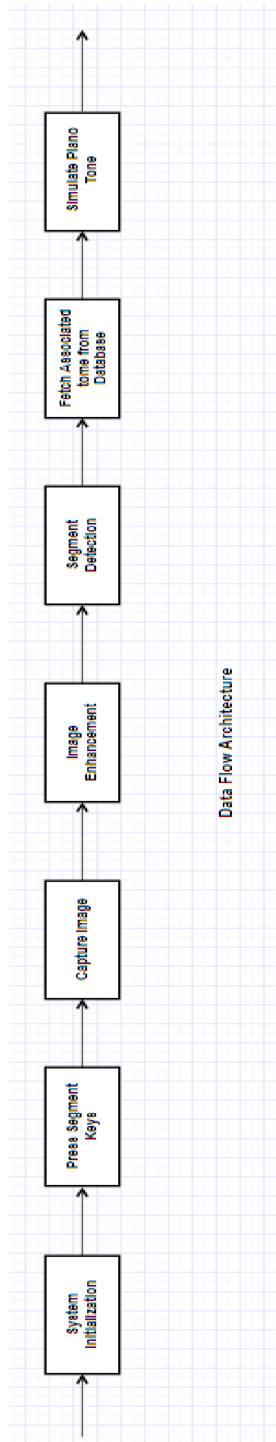

Figure 4.4: Data Flow Architecture

5. Context Diagram

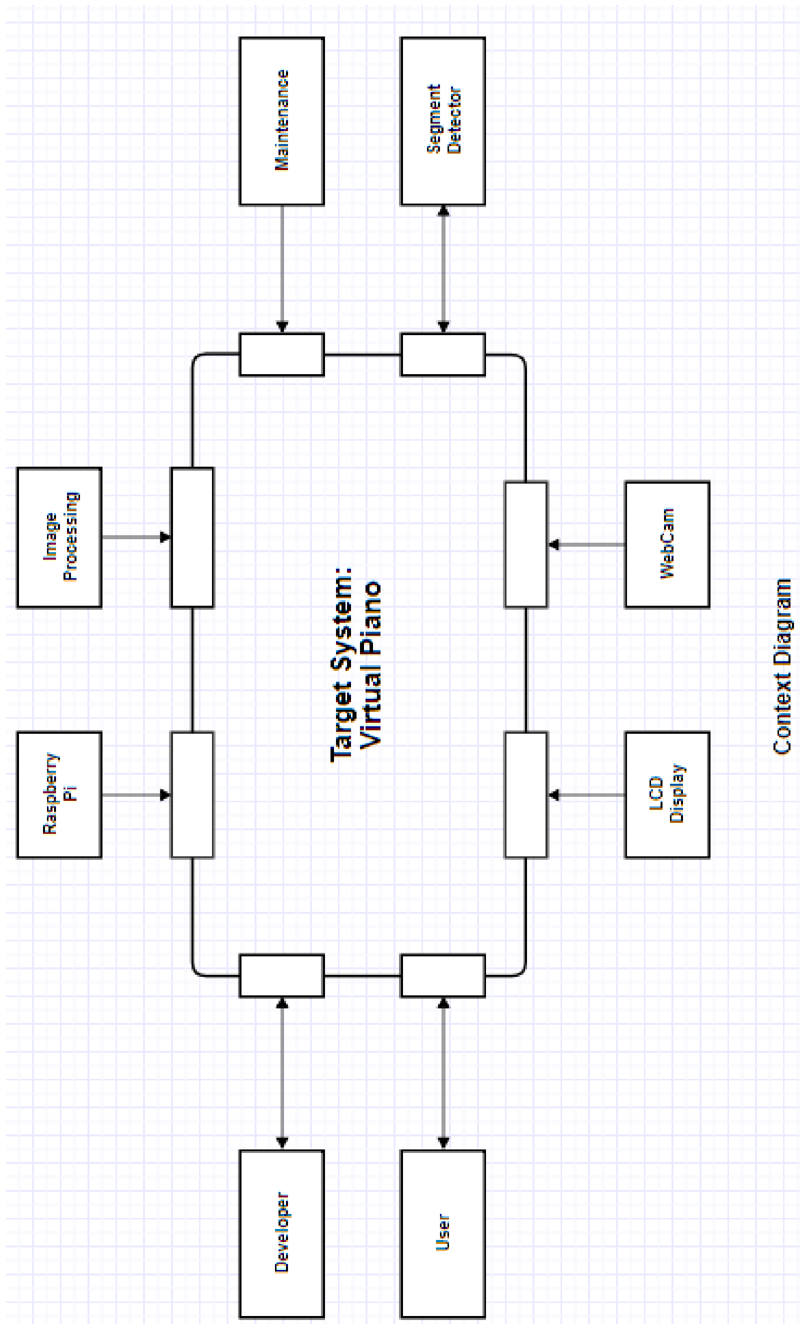

Figure 4.5: Context Diagram

4.2 Design Level Diagram

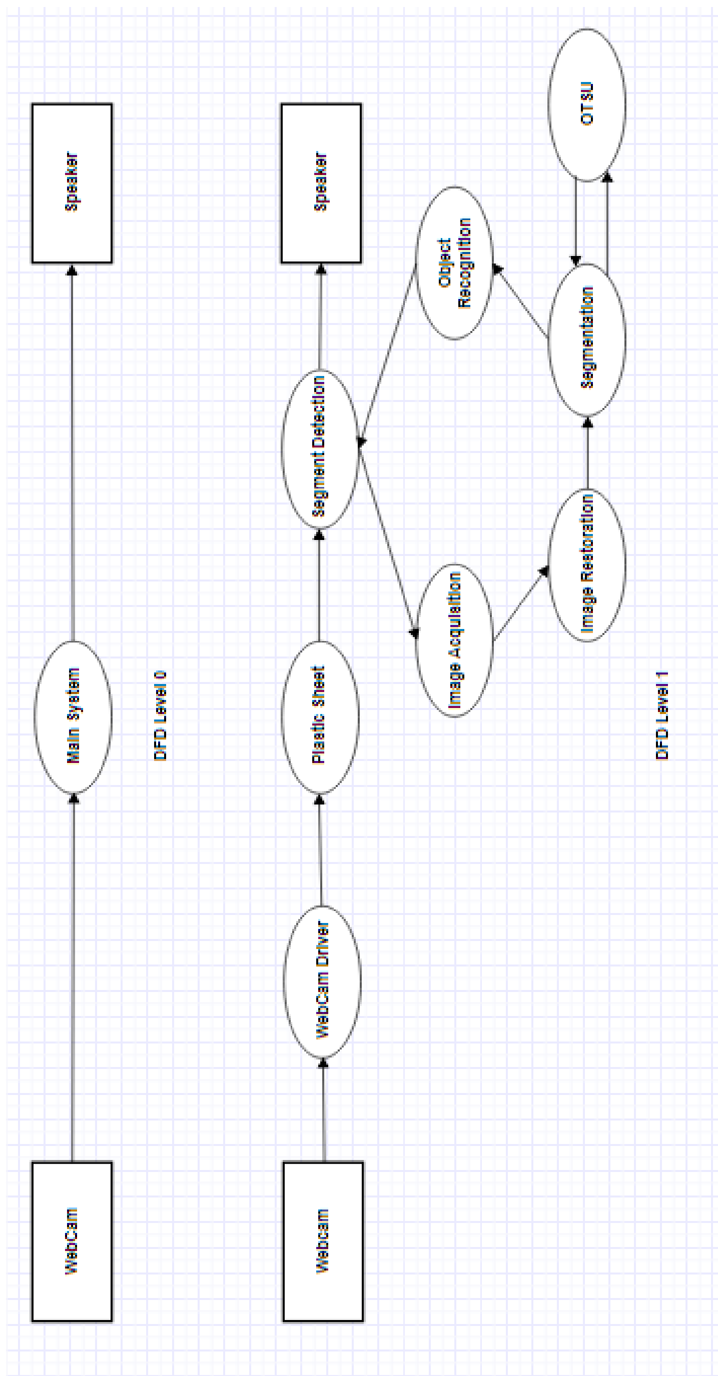

Figure 4.6: Design Level Diagram

4.3 User Interface Diagrams

1. Use case diagram

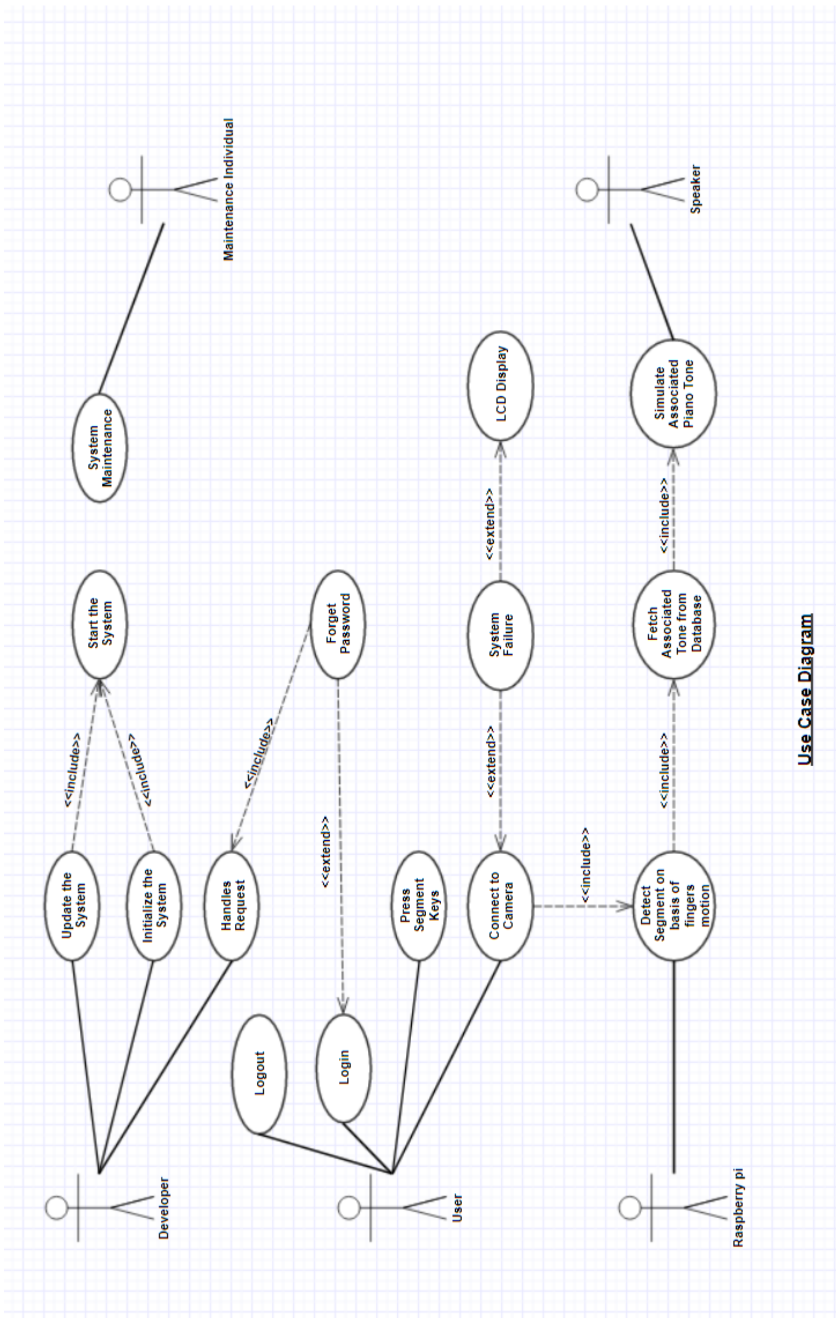

Figure 4.7: Use Case Diagram

2. Class Diagram

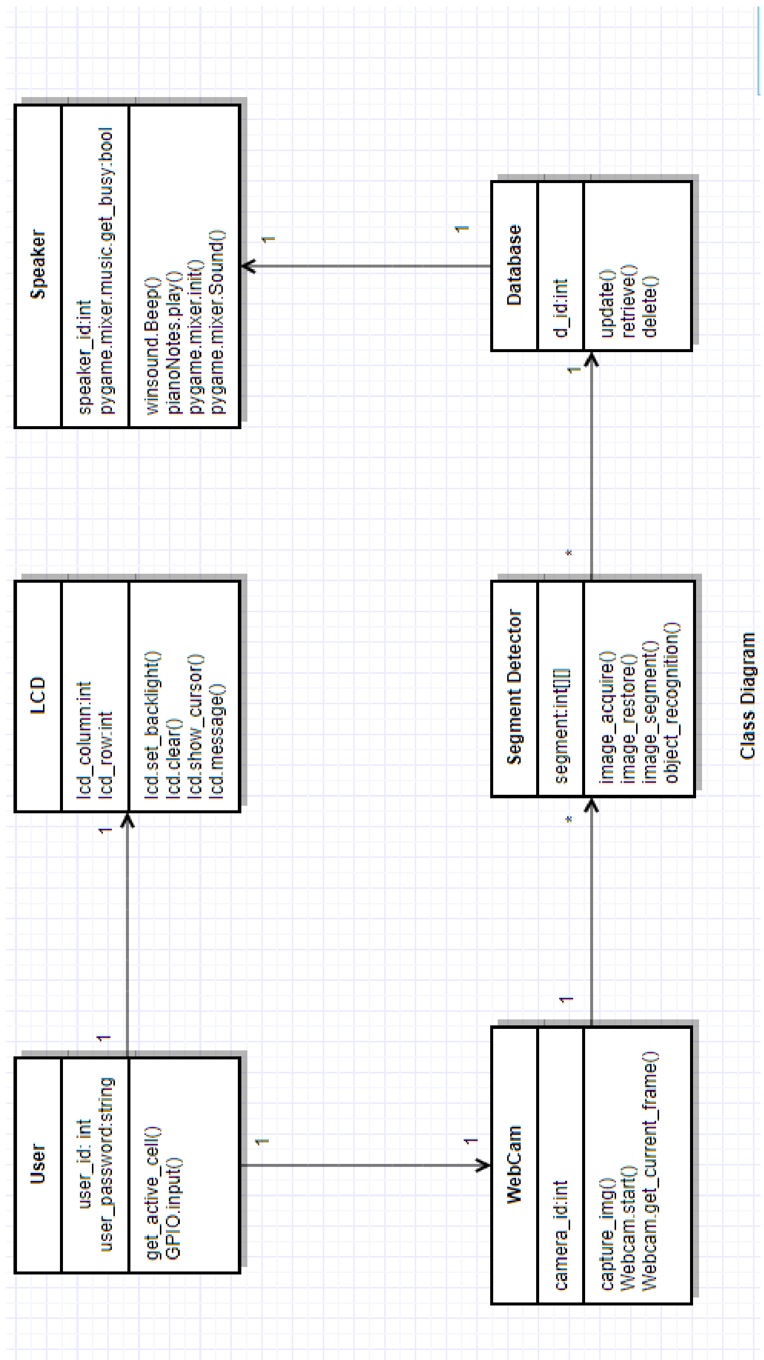

Figure 4.8: Class Diagram

3. Sequence Diagram

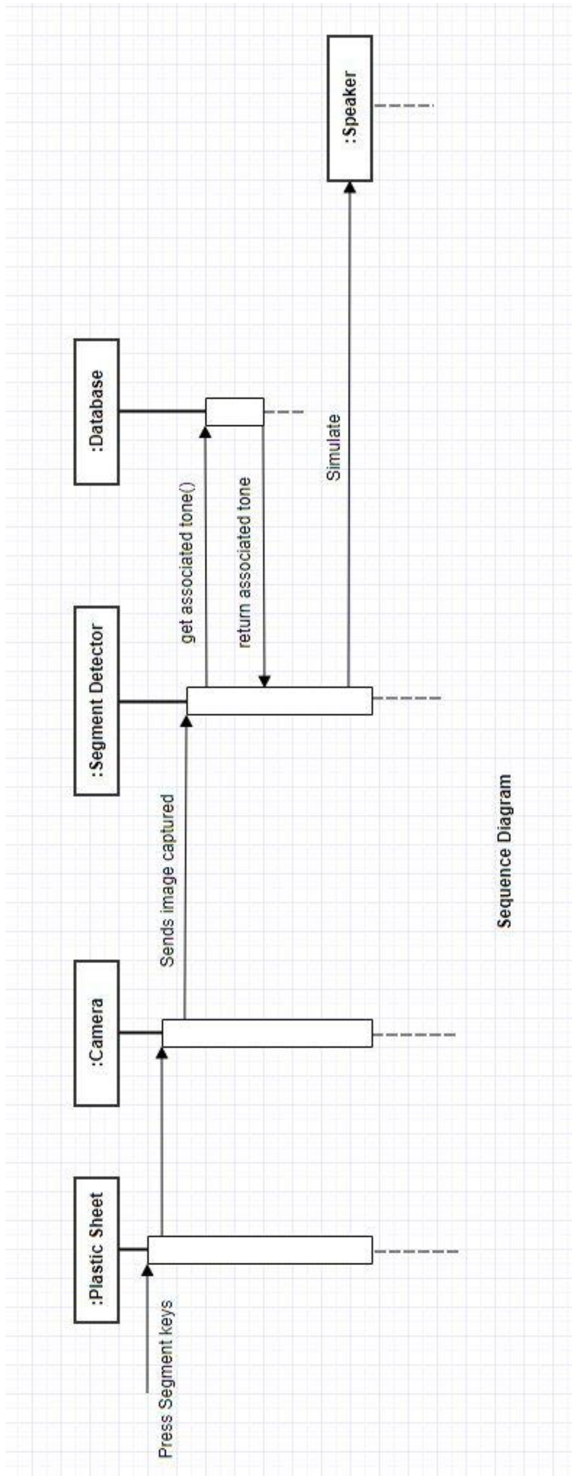

Figure 4.9: Sequence Diagram

4.4 System Screenshots

1. Semi-Transparent Plastic Sheet

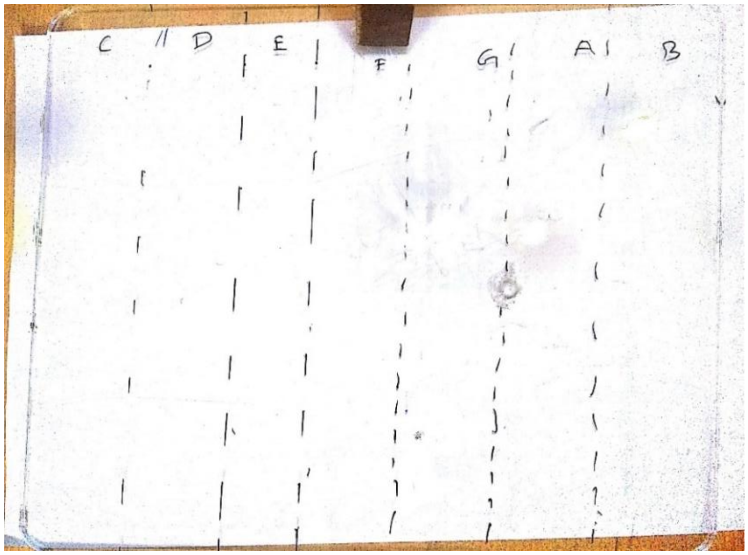

Figure 4.10: Semi-Transparent Plastic Sheet

2. Camera attached on wooden mount

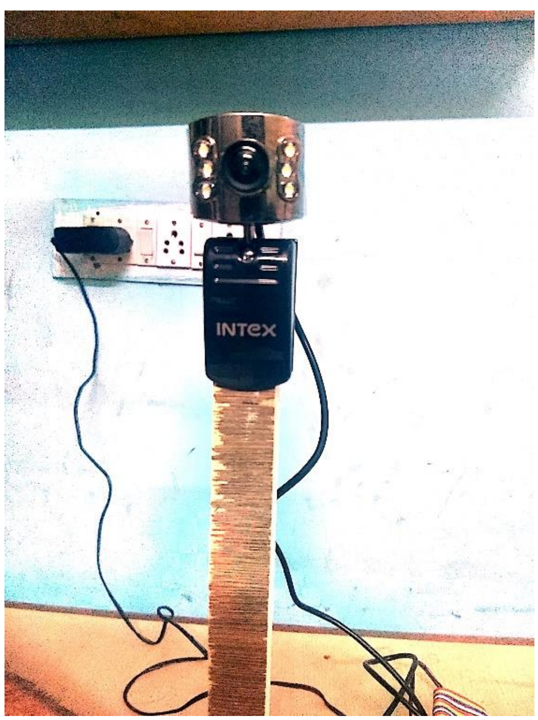

Figure 4.11: Camera attached on wooden mount

3. Speaker, Raspberry pi and LCD Display

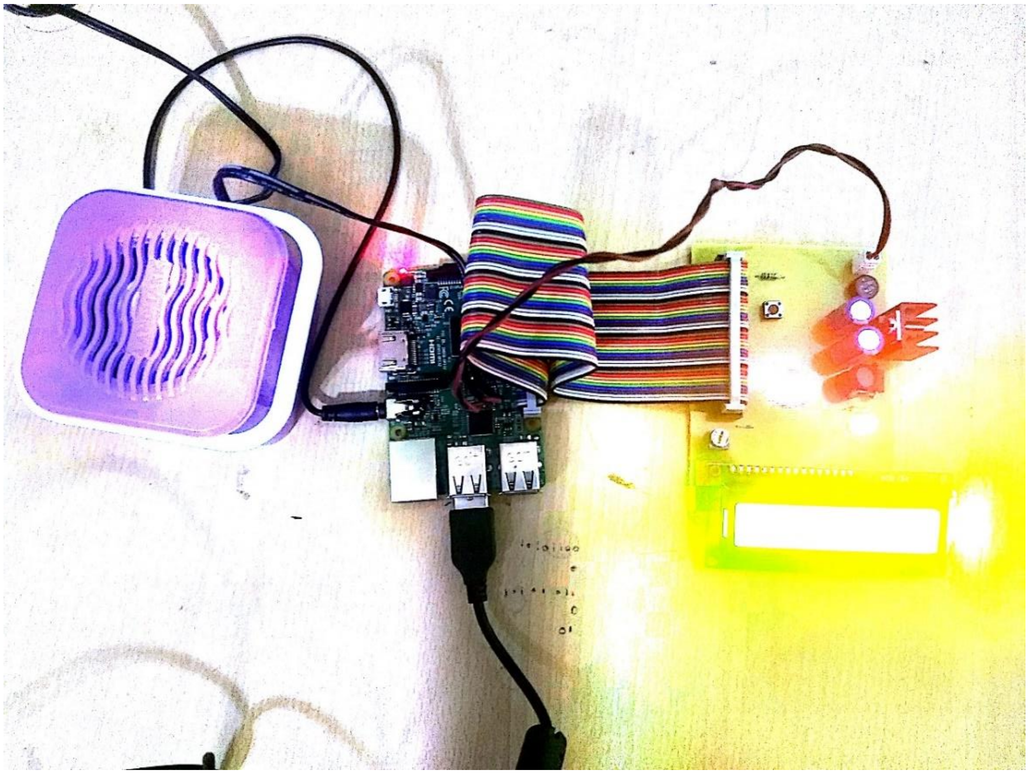

Figure 4.12: Speaker, Raspberry Pi and LCD Display

CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Work Accomplished

1. Finalized System Requirement Specification with finalized analysis model i.e. Use Case Diagrams, Class Diagrams, Data Flow Diagram and Sequence Diagram.
2. Finalized our design model consisting of Data Centered Architecture, Data Flow Architecture, Component Diagram and Context Diagram.
3. Soldering of components based on the circuit diagrams decided earlier.
4. Organized all the components based in the circuit diagram.
5. Finger detection and simulation of a piano tone.

5.2 Conclusion

This project is a great example of how the use of Raspberry Pi and Pi camera with Open CV's computer vision algorithms can do wonders in security standards. By compiling the latest version of Open CV, we got access to the latest and most interesting computer vision algorithms like finger recognition.

5.3 Social/Economic Benefits

1. We propose a portable virtual piano that just uses semi-transparent plastic sheet that can be carried anywhere and does not have any electronic components in it.
2. Large Pianos are prone to damage in travelling and these Virtual Pianos will take care of this issue.
3. Virtual Pianos will be much cheaper than traditional pianos.
4. It will help young children to get a feel for music and will act as a stepping stone for some of the world's greatest artists.

5.4 Reflections

1. Understood the workings of Raspberry Pi in detail.
2. Understood the implementation of Ada fruit in Python.

3. Understood the implementation of pygame library in Python.
4. Become proficient in working as an enthusiastic team

5.5 Future work plan

1. Code optimization to improve real time sensing.
2. Improved prototype with proper casing to give it a finishing and premium look.
3. Investigate using other real time sensing algorithms in OpenCV which might be more robust to varied lighting or other conditions.

REFERENCES

[1] A. Oka and M. Hashimoto, "Marker-less piano fingering recognition using sequential depth images", 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision, pp. 1-4, 2013.

[2] M. Sotirios and P. Georgios, "Computer vision method for pianist's fingers information retrieval," presented at the Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, Linz, Austria, pp. 5-8, 2008.

[3] C.-C. Lin and D. S.-M. Liu, "An intelligent virtual piano tutor," in Proceedings of the ACM international conference on Virtual reality continuum and its applications, Hong Kong, China, pp. 3-8, 2006.

[4] C. Deaker and R. Green, "A Computer Vision Method of Piano Tutoring, Without the Piano," Canterbury University, pp. 2-4, 2011.

[5] C.-H. Yeh, W.-Y. Tseng, J.-C. Bai, R.-N. Yeh, S.-C. Wang, and P.-Y. Sung, "Virtual Piano Design via Single-View Video Based on Multifinger Actions Recognition," the 3rd International Conference on Human-Centric Computing, pp. 1-5, 2010.

[6] F. Huang, Y. Zhou, Y. Yu, Z. Wang, and S. Du, "Piano AR: A Markerless Augmented Reality Based Piano Teaching System", International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 47-52, 2011.

[7] D. O. Gorodnichy and A. Yogeswaran, "Detection and tracking of pianist hands and fingers", the 3rd Canadian Conference on Computer and Robot Vision, pp. 62-63, 2006.

[8] K. Huang, E. Y. L. Do, and T. Starner, "PianoTouch: A wearable haptic piano instruction system for passive learning of piano skills, 12th IEEE International Symposium on Wearable Computers, pp. 41-44, 2008.

[9] Adam Goodwin, Richard Green, "Key detection for a virtual piano teacher", Image and Vision Computing New Zealand (IVCNZ) 28th International Conference of, pp. 282-287, 2013

[10] C. von Hardenberg, F. Berard, "Bare-hand human-computer interaction", Proceedings of the workshop on Perceptive user interfaces, pp. 1-8, 2001.

[11] Boga Vishal, K Deepak Lawrence, "Paper piano — Shadow analysis-based touch interaction", Man and Machine Interfacing (MAMI) 2nd International Conference on, pp. 1-6, 2017.

[12] Mohammad Akbari, Jie Liang, Howard Cheng, "A real-time system for online learning-based visual transcription of piano music", Multimedia Tools and Applications, vol. 77, pp. 25513, 2018.

[13] Chutisant Kerdvibulvech, Augmented Reality, Virtual Reality, and Computer Graphics, vol. 10325, pp. 251, 2017.

[14] D. Schwarz, S. Britton, R. Cahen, T. Goepfer. Musical Applications of Real-Time Corpus-Based Concatenative Synthesis. International Computer Music Conference (ICMC) (Copenhagen, Denmark), pp. 23, 2007.