VIQU-A SMART INTERVIEW/VIVA BOT

Capstone Project Report

End-Semester Evaluation

Submitted by:

(101503208) SHIVAM SHARMA

(101503211) SHOBHIT JAIN

(101503213) SHREYA AGGARWAL

BE Fourth Year, CSE

CPG No: 28

Under the Mentorship of

Dr. Maninder Kaur

Assistant Professor

Mr. Ashish Girdhar

Lecturer

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

Computer Science and Engineering Department
TIET, Patiala

NOV 2018

ABSTRACT

In the various laboratory classes in the college there have numerous viva(s) and also go through various technical interviews in the process of campus recruitment. This task of conducting viva(s) and interviews is quite cumbersome for the teachers and interviewers as it takes a lot of preparation. Also, the checking process is quite redundant, time taking and tiring which leads to delayed results. Moreover, at the time of every viva/interview teacher/interviewer has to find the same set of questions which is very time consuming and redundant.

The came up with an innovative solution to the above-mentioned problems. ViQu, an interview/viva taking bot which can handle the situation such as over-intimidating interviews/viva(s) and speed up the process in various scenarios.

The aim to create a hardware bot based on Raspberry Pi 3, single-board computer. It will ask students/candidates questions of categories – fill in the blanks, true/false, multiple choice questions and one-word answers. It will be attached with some sensors such as mic (for audio input from the candidate), speakers (for audio output from the ViQu), etc. and mark sheet for the analysis purposes of the teacher and the student/candidate will be generated.

DECLARATION

We hereby declare that the design principles and working prototype model of the project entitled “ViQu- A smart Interview/viva bot” is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of Dr. Maninder Kaur and Mr. Ashish Girdhar during 6th and 7th semesters (2018).

Date: November 19, 2018

Roll No.	Name	Signature
101503208	Shivam Sharma	
101503211	Shobhit Jain	
101503213	Shreya Aggarwal	

Counter Signed By:

Faculty Mentor:

Dr. Maninder Kaur
Assistant Professor
CSED,
TIET, Patiala

Mr. Ashish Girdhar
Lecturer
CSED,
TIET, Patiala

ACKNOWLEDGEMENT

We would like to express our thanks to our mentor Dr. Maninder Kaur & Mr. Ashish Girdhar. They have been of great help in our venture, and an indispensable resource of technical knowledge. They are truly amazing mentors 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: November 19, 2018

Roll No.	Name	Signature
101503208	Shivam Sharma	
101503211	Shobhit Jain	
101503213	Shreya Aggarwal	

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Program Interface
ASR	Automatic Speech Recognition
GTTS	Google Text-to-Speech
IOT	Internet of Things
ML	Machine Learning
NLP	Natural Language Processing
OS	Operating System
PC	Personal Computer
STT	Speech-to-Text
TTS	Text-to-Speech

INTRODUCTION

1.1 Project Overview

In the present world various tasks are being automated owing to one reason or another. The tasks maybe redundant in nature or maybe the bots can be more efficient in doing the job or a completely different reason.

One such task is to create a bot foe technical interviews/viva(s) which ought to be just talent-based exams rather than being an intimidating round of questions. So, to remove the intimidating factor of a face to face interview/viva and also to make the process easier for the interviewer a bot has been created.

To accomplish the given task, it was needed to combine the capabilities of various different resources such as the different speech engines available such as, google speech engine, espeak – a speech engine for linux distros, the hardware capabilities being borrowed from the Raspberry Pi 3 mini credit card sized computer and a few other libraries available in python which helper accomplish some of the tasks such creation of the marking and text pattern matching modules.

Firstly, the most important component being used is the Raspberry Pi 3 which acts as the base to all the software being run and helps in creating a stand-alone product finally. Raspberry Pi a mini credit card sized computer with Wi-Fi, Bluetooth, Ethernet, etc. capabilities. This device acts as the host to all the software and the service engines working in conjunction to provide the bot needed. The espeak speech engine and the google speech engine have been used to give the bot to gain the text-to-speech and speech-to-text services. These services have been used via the help of some wrapper libraries available in python such as talkey and speech-recognition, which have various functions and setup classes for the developers to obtain the desired results. The marking module is there to mark the answers given by the student according to certain standards, this module has a helper module named pattern which is there to aid the matching of keywords provided. Here the fuzzywuzzy logic was used to match the keywords even if they have a partial match ratio which may eventually lead to a correct answer. Finally, this bot helps in conducting the interview/viva of a number of candidates and can then create a spreadsheet containing the marks of students against their roll numbers.

1.1.1Technical Terminology

Various technical terms used in the project are as follows:

I. Speech Recognition

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as automatic speech recognition, computer speech recognition or speech to text.

II. Speech Synthesis

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products.

III. Natural Language Processing

Natural language processing is a subfield of computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human languages, in particular how to program computers to process and analyse large amounts of natural language data.

IV. Internet of Things

The Internet of things is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect, collect and exchange data.

V. Raspberry Pi

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries.

VI. Raspbian

Raspbian is a Debian-based computer operating system for Raspberry Pi. There are several versions of Raspbian including Raspbian Stretch and Raspbian Jessie. Since 2015 it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the family of Raspberry Pi single-board computers.

1.1.2 Problem Statement

In the various laboratory classes in the college there have numerous viva(s) and also go through various technical interviews in the process of campus recruitment. This task of conducting viva(s) and interviews is quite cumbersome for the teachers and interviewers as it takes a lot of preparation. Also, the checking process is quite redundant, time taking and tiring which leads to delayed results. Moreover, at the time of every viva/interview teacher/interviewer has to find the same set of questions which is very time consuming and redundant.

1.1.3 Goals

Our main goal is to help teachers/interviewers in the process of conducting viva(s)/technical interviews by simplifying the whole process of asking questions, evaluating answers and creating the final result sheet. The team want to simulate the real-world viva/interview scenario as well as possible. Also, to create a question bank of questions with different difficulty levels to help us in the future.

1.1.4 Solution

Creating a product “ViQu” which is a proposed solution to the problem. A product based on Raspberry Pi 3 along with some sensors will be created, which will be able to conduct viva(s) and technical interviews and even mark them hence, creating a digitized result sheet. It will ask students/candidates questions of categories – fill in the blanks, true/false, multiple choice questions and one-word answers.

Moreover, a database of viva-like questions will be created overtime by the teachers which will help in the future. To create the real-world scenario, the question bank will have questions of 3 difficulty levels – easy, medium and hard and each student/candidate will be asked questions of different types according to the teacher's/interviewer's needs.

Finally, the product ViQu will be able to conduct interviews/viva(s), mark them, and a fine database of questions will be created for the above.

1.2 Need Analysis

In the various laboratory classes in college the team have numerous viva(s) and go through various technical interviews in the process of campus recruitment. This task of conducting viva and interviews is quite cumbersome for the teachers and interviewers as it takes a lot of preparation. The system being proposed is to simplify the process of interview/viva.

I. Need to simplify the Interview/viva process

As the number of students is increasing every year the work load on teachers also keeps increasing. Viva(s) are an integral part of lab work but on the other hand they are too tedious for the teachers. Hence, ViQu will take this responsibility off their shoulders. The process of interviews is redundant and the interviewer can get tired of asking same set of questions which impacts the interview of the candidate. ViQu can take this responsibility.

II. Need for a question bank

Now-a-days questions need to be created by interviewers/teachers every time there is an interview/viva. But with ViQu a large database of subjective questions of categories – fill in the blanks, true/false, multiple choice questions and one-word answers will be created overtime which will help in the future.

1.3 Research Gaps

The research found two AI based video interview bots made till date, used for the purpose of recruitment by organizations. The main problem which was found with these interview bots is that they are software based and require subscription which make them very costly. They are not available as hardware. Due to this they cannot be used for general purpose and also everyone cannot afford to use these interview bots. These bots cannot be used in college environment as they may prove to be cost inefficient. Also, it was found that there was no question bank created, which could help in the future to make the interview bot standalone and the recruiter need not provide the list of questions, which may be same to some extent, again and again.

1.4 Problem definition and Scope

The problem as explained above is to simplify the viva/interview process as it gets very tedious for both the interviewer and the candidate itself. Moreover, the question set needs to be created again and again each time an interview needs to be conducted.

Hence, the scope of this problem is to create a hardware bot which will be able to conduct interview/viva (mainly technical) by asking questions of categories – fill in the blanks, true/false, multiple choice questions and one-word answers and eventually create a large question bank of similar question set.

1.5 Assumptions and Constraints

The proposed system has a certain constraints and assumptions which are needed for the project to run as mentioned below.

I. Assumptions

TABLE 1: Assumptions

S. No.	Assumptions
1	The prerequisites to start an interview/viva • A csv (comma delimited) file named "student.csv" should be provided with student roll numbers under a single column to be stored in the folder /home/viqu/data/ (Sample provided). • A file named "questions.txt" containing the questions (each question in a single line) to be provided and stored in the folder /home/viqu/data/ (Sample provided). • A file named "keywords.txt" containing the corresponding keywords (keywords of each question in a single line, and separated by commas) to be provided and stored in the folder /home/viqu/data/ (Sample provided) • Setup the configuration file "conf.txt" which contains the number of easy, medium and hard questions (each in separate line) to be stored in the folder /home/viqu/data/ (Sample provided)
2	Obtaining final result • The final answers given by each student will be stored in the folder /home/viqu/data/studentAnswers/ in the format <roll number>_answers.txt. This folder must be emptied after the process to create space for further usage of the bot. • The final spreadsheet with student roll numbers and marks is stored in the folder /home/viqu/data/ as student.csv, which must also be copied into the invigilator's system after the process to prepare the bot for next process.

II. Constraints

TABLE 2: Constraints

S. No.	Constraints
1	The proper file tree system (as provided initially) should not be disturbed for the proper functioning of the bot.
2	The interview/viva environment should be noise free.
3	Questions can only be of categories – fill in the blanks, true/false, multiple choice questions and one-word answers.
4	The candidate must speak clear and grammatically correct English.
5	The bot should be constantly connected to the internet either via WiFi or ethernet.

1.6 Approved Objectives

In order to fulfill our capstone project, it is intended to achieve the following objectives:

I. Create a hardware-based bot which can conduct an interview/viva

II. To ask questions via TTS: text-to-speech

III. To receive answers via STT: speech-to-text

IV. To mark the answers based upon the keywords to the questions provided

V. To create a final marksheet with student roll number and marks

VI. To provide the answers of each candidate in digital format to the invigilator

VII. Eventually create a question bank of relevant questions

VIII. To improve our skill set

1.7 Methodology Used

I. Raspberry Pi 3 is being used for the purpose of creating the stand-alone bot which will be conducting the viva/interview.

II. Various peripherals used such as speakers and microphone.

III. All the basic coding is done in Python 2.7.

IV. Various python libraries are used to perform tasks such as STT, TTS, etc.

V. Text-to-speech (TTS)

The library used here is “gTTS” which is a google text-to-speech library and “talkey” which is a wrapper to the espeak speech engine for linux distros. The text to be converted to speech is sent to the engine and a corresponding audio file is received.

VI. Speech-to-text (STT)

The library used here is “speech_recognition 3.8.1”. It contains various speech recognition engines which were tested for accuracy and the one provided by Google Inc (free version)/Google Cloud Services (paid version), was found to be the most accurate amongst all of them. The microphone connected to the device is being used as the audio source which is used as a recording medium. This audio is then sent to the speech recognition engine and a text is received from the API.

VII. Keyword matching

The libraries used here are “fuzzywuzzy”, “nltk” and “re”. Keyword matching is done via basic n-gram model which just does a fuzzy string comparison. The nltk and re libraries is used to create n-grams of the textual answer. The fuzzywuzzy library has a function fuzz.ratio() which gives us the matching ratio between two strings i.e. an approximate percentage match between two strings is obtained.

VIII. Number of libraries are used here such as

“time” is used to create required delays in the system process. “os” is used to perform operating system related tasks. The python standard library “shutil” works as a helper in various tasks. The “csv” library is used to handle .csv files.

1.8 Project Outcomes and Deliverables

A raspberry Pi 3 based hardware bot is created which fully capable of conducting an interview/viva of a group of individuals and can generate a marksheet of these candidates and then also transcribe the answers given by the individuals which are then provided to the invigilator for further analysis.

Deliverables are -

I. Hardware bot

II. Project documentation with detailed discussion of the methodology and procedures used

1.9 Novelty of Work

As observed in the research, very few such bots have been created which are capable of the same task. Moreover, they are not standalone systems and need a fully capable computer system to work and are very costly and account for a subscription plan. However, ViQu is a stand-alone system which has a one-time cost and works with considerable efficiency without any subscription required.

2.1 Literature Survey

Study of already existing systems is very crucial to the creation of such a large system as leads to the knowledge of already existing systems and tells us the methodology which can be adopted. Following are the findings of the study.

2.1.1 Theory Associated with Problem area

Throughout the college and campus recruitment candidates go through numerous viva(s) and various technical interviews. This task of conducting viva(s) and interviews is quite cumbersome for the teachers and interviewers as it takes a lot of effort and preparation. Also, the checking process is quite redundant, time taking and tiring which leads to delayed results. Moreover, at the time of every viva/interview teacher/interviewer has to find the same set of questions which is very time consuming and redundant.

2.1.2 Existing Systems and Solutions

Organizations conduct interviews for recruitment and other purposes and also there are numerous viva(s) conducted during college. Artificial Intelligence (AI) has made many of our jobs easier and simpler. Now AI can help in the process of interviews which is otherwise cumbersome and redundant.

Gecko is an AI-based Video Interview Bot. It aids large-scale recruitment, with reduced human intervention errors. It Speeds Up the Interview Process and shortens time to recruit. Recruiters can set up interviews with custom questions that require video responses, which can later be played back for detailed analysis and review. It uses NLP and sentiment analysis to converse, produce reports and provide insights.

Impress is another AI powered recruiting solution. It automates the hiring process, helps choose the best candidate and in lesser amount of time. Based on Machine Learning, it uses Natural Language Processing (NLP) and deep learning to efficiently interact with the candidates. It takes video interviews. After the interview is completed, results are evaluated by a model which learns from the choices of the recruiters through ML. Recruiters get to choose which assessments to evaluate candidate with, whether that's behavioral and cognitive skills, technical skills, competencies, or cultural fit, i.e. it is easily compatible to recruiter's needs. Also, the chatbot evaluates the candidate's response before deciding if they move to the next round.

2.1.3 Research Findings for Existing Literature

TABLE 3: Research Findings for Existing Literature

S. No.	**Roll Number**	**Name**	**Paper Title**	**Tools/ Technology**	**Findings**	**Citation**
1	101503208	Shivam Sharma	Comparative Analysis of Different Operating Systems for a Raspberry Pi	Raspberry Pi, various operating systems for Raspberry Pi	Raspbian was found to be the best OS for Raspberry Pi because of its support and availability of numerous linux as well as python packages.	Nimat *et. al.* [1]
2	101503208	Shivam Sharma	Smart Speaker using Raspberry Pi	Raspberry pi, text-to-speech, speech-to-text	The work aims at the development of a personal voice assistant which can assist users using speech commands. The Raspberry Pi collects speech input and interprets it to manage certain tasks.	Jambagi *et. al.* [2]
3	101503208	Shivam Sharma	Gecko – AI bots for recruitment	AI interview bot, NLP, sentiment analysis	Gecko is an AI-based Video Interview Bot. It uses NLP and sentiment analysis to converse, produce reports and provide insights.	Jhunjhunwala [3]
4	101503208	Shivam Sharma	Cloud Text-to-Speech	Text-to-Speech (gTTS), Machine Learning	Usage of Google Text-to-speech conversion API. Cloud Text-to-Speech API converts text or Speech Synthesis Markup Language (SSML) input into audio data of natural human speech.	Google Inc. [4]
5	101503208	Shivam Sharma	Cloud Speech-to-Text	Speech-to-text	Cloud Speech-to-Text enables easy integration of Google speech recognition technologies into developer applications.	Google Inc. [5]
6	101503208	Shivam Sharma	Comparing Open-Source	Speech Recognition,	Kaldi toolkit gave the best performance, even better than Sphinx.	Matarneh *et.al.* [6]

			Speech Recognition Toolkits	ASR systems		
7	101503211	Shobhit Jain	impress	AI powered recruiting solution, Machine Learning, NLP, deep learning	It takes video interviews. After the interview is completed, results are evaluated by a model which learns from the choices of the recruiters through ML.	Ahuja *et. al.* [7]
8	101503211	Shobhit Jain	Natural language processing	Natural Language Processing, Artificial Intelligence	This paper gives an overview of the current trends in NLP and discuss the possible applications of traditional AI techniques and their combination.	Gelbukh [8]
9	101503211	Shobhit Jain	The Future Internet of Things: Secure, Efficient, and Model-Based	Internet of Things, cloud, various Standards	The paper proposes a solution jointly addressing security, efficiency, privacy, and scalability needed to support continued expansion in IoT. A solution modeled on human use of context and cognition, leveraging cloud resources to facilitate IoT on constrained devices is proposed.	Siegel *et. al.* [9]
10	101503211	Shobhit Jain	Raspbian	Raspbian OS for Raspberry Pi	Raspbian is a free and recommended operating system for Raspberry Pi. It comes with over 35,000 packages and helps runs various modules on the Raspberry Pi.	Hughes *et. al.* [10]
11	101503211	Shobhit Jain	Speech Recognition System – A Review	Speech Recognition, Speech Processing	Speech Recognition System is growing day by day and has unlimited applications. The study has shown the overview of the speech recognition process, its basic model, and applications.	Naziya and Deshmukh [11]
12	101503211	Shobhit Jain	Internet of Things Applications, Challenges and Related Future Technologies	Internet of Things	This paper explains the concept of many IoT applications and future possibilities for new related technologies in addition to the challenges that facing	Aldein *et. al.* [12]

					the implementation of the IoT.	
13	101503213	Shreya Aggarwal	Natural Language Processing: A Review	Natural Language Processing, string matching, keyword search	It explains technologies, such as string matching, keyword search etc.	Joseph *et. al.* [13]
14	101503213	Shreya Aggarwal	NLTK 3.3	n-grams	The Natural Language Toolkit (NLTK) is an open source Python library for Natural Language Processing. It specifies on the usage of n-grams in python.	Bird *et. al.* [14]
15	101503213	Shreya Aggarwal	Smart Interviews: AI-Powered Recruitment	AI based interviews, speech recognition, sentiment analysis	After the interview, an email with a report including the contact data, a sentiment analysis process, a summary, and the transcription is sent to the human recruiter.	Francis [15]
16	101503213	Shreya Aggarwal	Scope of IoT: Performance and Hardware Analysis Between Raspberry Pi-3 And Arduino Uno	Raspberry Pi 3, Arduino Uno	After analyzing parameters like RAM, Processor, OS,GPIO pins, Cost, Power & energy consumption and adaptability to cloud computing, it is observed that Raspberry Pi 3 is more dynamic, user friendly and adaptable to latest technologies.	Irshad and Feroz [16]
17	101503213	Shreya Aggarwal	Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)	Speech Recognition	After calculating word error rate(WER), Google API was found to be superior with WER of 9% whereas Microsoft API had 18% and Sphinx had 21% WER.	Këpuska and Bohouta [17]
18	101503213	Shreya Aggarwal	A partial ratio and ratio-based fuzzy-wuzzy procedure for characteristic mining of mathematical formulas from documents	Sequence matcher, Levenshtein Distance and Fuzzy-Wuzzy, efficiency measure, sensitivity analysis	Fuzzy-Wuzzy partial ratio algorithm scored better over the other variants on efficiency measure and sensitivity analysis.	Rao *et. al.* [18]

2.1.4 The Problem that has been Identified

There have been two interview bots made till date. They both are AI based video interview bots for the purpose of recruitment by organizations. The main problem with these interview bots is that they are software based and require subscription which make them very costly. They are not available as hardware. Due to this reason they cannot be used for general purpose and also everyone cannot afford to use these interview bots. These bots cannot be used in college environment as they may prove to be cost inefficient.

2.1.5 Survey of Tools and Technologies used

Speech Recognition is commonly employed in everyday applications. Due to the increasing number of Automated Speech Recognition (ASR) systems, such as Microsoft, Google, Sphinx, WUW, HTK and Dragon, it becomes very difficult to know which of them are needed. The idea of this paper is to design a tool that will be used to test and compare commercial speech recognition systems, such as Microsoft Speech API and Google Speech API, with open source speech recognition systems such as Sphinx. After calculating word error rate (WER), it was observed that the Google API is the best [17].

Retrieval of text from data is a key predicament in present circumstances. To achieve this, three different algorithms viz., Sequence matcher, Levenshtein Distance and Fuzzy-Wuzzy, were considered. Two different variants of Fuzzy-Wuzzy are found applicable to this study out of four variants. Performance of these variants in retrieving mathematical texts, is calculated using efficiency measure, sensitivity analysis and time series exploration. Fuzzy-Wuzzy partial ratio algorithm scored better over the other variants on efficiency measure and sensitivity analysis.It compares three different algorithms Sequence matcher, Levenshtein Distance and Fuzzy-Wuzzy (two different variants). Performance of these variants in retrieving mathematical texts, is calculated using efficiency measure, sensitivity analysis and time series exploration. Fuzzy-Wuzzy partial ratio algorithm scored better over the other variants on efficiency measure and sensitivity analysis [18].

Hiring processes, when conducted by humans at least, have always been problematic because bias is so often unconscious. Personality and psychometric testing, blind auditions, webcam interviews and nameless CVs are on the rise, but in a face-to-face environment, anything from gender, race, clothing, education and accent can provide an unwitting platform for discrimination. Humans are inconsistent where robots are incapable of being anything but consistent [15].

Speech synthesis is artificial simulation of human speech with by a computer or other device. The counterpart of the voice recognition, speech synthesis is mostly used for translating text information into audio information and in applications such as voice-enabled services and mobile applications. Apart from this, it is also used in assistive technology for helping vision-impaired individuals in reading text content. With the advent of various efficient speech engines, several options were available for deployment such as espeak for linux distros and google cloud-based speech engine which are primarily used in the project [19].

2.2 STANDARDS

Various standards used in the project are as follows:

I. IEEE 802.11b

It is a set of media access control (MAC) and physical layer (PHY) specifications for implementing wireless local area network (WLAN) computer communication. They are the world's most widely used wireless computer networking standards, used in most home and office networks to allow laptops, printers, and smartphones to talk to each other and access the Internet without connecting wires.

The 802.11b standard has a maximum raw data rate of 11 Mbit/s, and uses the same media access method defined in the original standard. This standard uses DSSS modulation. Devices using 802.11b experience interference from other products operating in the 2.4 GHz band.

II. Gazette Notification No.G.S.R. 45(E) dated 28-01-2005

Standard used to regulate the Raspberry Pi 3 model B as per the provisions mentioned in the above notification dated 28-01-2005 as amended from time to time.

III. UL94

UL 94, the Standard for Safety of Flammability of Plastic Materials for Parts in Devices and Appliances testing, is a plastics flammability standard released by Underwriters 6 may not burn with flaming combustion for more than 10 seconds after either application of the test flame. The total flaming combustion time may not exceed 50 seconds for the 10 flame applications for each set of 5 specimens. The specimens may not burn with flaming or glowing combustion up to the holding clamp. The specimens may not drip flaming particles that ignite the dry absorbent surgical cotton located 300 mm below the test specimen. The specimens may not have glowing combustion that persists for more than 30 seconds after the second removal of the test flame.

2.3 Software Requirements Specification

The software requirement specifications of the proposed system are as follows.

2.3.1 Introduction

A product based on Raspberry Pi 3 along with some sensors is being created, which will be able to conduct viva(s) and technical interviews and create a digitized result sheet.

2.3.1.1 Purpose

The purpose of this document is to provide a detailed overview of our hardware bot which conducts interviews/viva(s), its parameters and its goals. This document specifies various functional and non-functional requirements in details and it also specifies which of these requirements is satisfied by which feature of the software. It also describes various constraints and standards applicable in software of this domain. It includes description of all software/hardware used.

2.3.1.2 Intended Audience and Reading Suggestions

The technical interview/viva process is a task which requires a large man power and can get a bit intimidating and redundant. The bodies which look forward to conducting the interviews/viva(s) are the prime audience of the bot being created. This product will help them to simplify the process. As the product being created has a lot of future scope, this project can also be studied by the students or developer teams so as that in future the product can reach its true potential. Certain suggestions do come with the bot such as the questionnaire creation should be done as per the guidelines and the hardware must be handled properly.

2.3.1.3 Project Scope

The problem as explained above is to simplify the interview/viva process as it gets very tedious for both the interviewer and the candidate itself. Moreover, the question set needs to be created again and again each time an interview needs to be conducted. Hence, the scope of this project is to create a hardware bot which will be able to conduct interview/viva (mainly technical) consisting of questions of category – fill in the blanks, true/false, multiple choice questions and one-word answers, and eventually create a large question bank of similar question set.

2.3.2 Overall Description

The problems faced in today's interview/viva processes are trivial and need to be solved. So, a system is being proposed to do the same as explained below.

2.3.2.1 Product Perspective

Proposed system to create a product "ViQu", a product based on Raspberry Pi 3 along with some sensors will be created, which will be able to conduct viva(s) and technical interviews consisting of questions of categories – fill in the blanks, true/false, multiple choice questions and one-word answers, and even mark them hence, creatin a digitized result sheet.

Moreover, a database of similar questions will be created overtime by the teachers which will help in the future. To create the real-world scenario, the question bank will have questions of 3 difficulty levels – easy, medium and hard and each student/candidate will be asked questions of different types according to the teacher's/interviewer's needs. Finally, the product ViQu will be able to conduct interviews/viva(s), mark them, and a fine database of questions will be created.

2.3.2.2 Product Features

The product features are as follows:

I. Create a hardware-based bot which can conduct an interview/viva

II. To ask questions via TTS: text-to-speech
III. To receive answers via STT: speech-to-text
IV. To mark the answers based upon the keywords to the questions provided
V. To create a final marksheet with student roll number and marks
VI. To provide the answers of each candidate in digital format to the invigilator
VII. Eventually create a question bank of relevant questions

2.3.3 External Interface Requirements

The external interface requirements are divided into following parts namely user interface, hardware interface and software interface.

2.3.3.1 User Interfaces

The proposed system will run on MobaXterm and will display appropriate instructions on-screen. These software tools provide for a system to connect the Raspberry Pi 3 model B to the PC via ethernet connection.

2.3.3.2 Hardware Interfaces

In the proposed system the user will only interact with the microphone and the speakers/earphones which act the input and output devices respectively to the hardware system.

2.3.3.3 Software Interfaces

Software used are MobaXterm and Angry IP Scanner for the Raspberry Pi setup.

2.3.4 Other Non-functional Requirements

The requirements related to the proper functioning of the system as far as the external conditions are required are mentioned below.

2.3.4.1 Performance Requirements

The marking of the answers is totally dependent on the transcription accuracy so a good microphone along with a decent running internet connection is highly important for the performance of the proposed system.

2.3.4.2 Safety Requirements

UL 94, the Standard for Safety of Flammability of Plastic Materials for Parts in Devices and Appliances testing, is followed. The device must be supplied appropriate amount of power. Very high or very low power can damage the bot. Also it must not be kept in a closed container as it will heat up and bot will be damaged. For the same reason at one time it can take up to 30 interviews/viva(s).

2.3.4.3 Security Requirements

Since the device is always in the control of examiner, the students can never temper with it, making it safe. Also, to make it more secure a feature will be provided where the final spreadsheet of marks can only be open with the help of a password, further making the data more secure.

2.4 Cost Analysis

Since it is a hardware-based project, various devices must be bought to create it.
The budget related to these devices is as follows

TABLE 4: Cost Analysis

S. No.	**Product**	**Cost in INR**
1	Raspberry Pi 3	3199
2	Mic	449
3	Speakers	850
4	Ethernet Wire	200
5	HDMI Cable	349
6	SD card (16 GB)	499
	Total	**5546**

2.5 Risk Analysis

Risks involved are: the device will need some noticeable amount of time to start the interview process. Another risk factor that has been considered is that it can only take interview/viva of around 30 candidates/students at one time as it may heat up and would get destroyed. Another risk is that it needs an appropriate amount of power supply. High and low power supply can damage the interview/viva bot.

METHODOLOGY ADOPTED

3.1 Investigative Techniques

The proposed project is experimental as the work being done in this project has not been done before. The technologies being taken up are some of the latest ones such as natural language processing, fuzzy string matching, etc.

The proposed system is quite new as there no such solution already existing in the market. Although some bots are there for the same but all of these are software based and have a hefty subscription fee associated to them.

Moreover, such an endeavor of connecting the interview/viva process to an IOT device has not been done before.

3.2 Proposed Solution

Our team is trying to create a product "ViQu", a product based on Raspberry Pi 3 along with some sensors will be created, which will be able to conduct viva(s) and technical interviews comprising of questions of categories – fill in the blanks, true/false, multiple choice questions and one-word answers, and even mark them hence, creating a digitized result sheet.

Moreover, a database of viva-like questions will be created overtime by the teachers which will help in the future. To create the real-world scenario, the question bank will have questions of 3 difficulty levels – easy, medium and hard and each student/candidate will be asked questions of different types according to the teacher's/interviewer's needs.

Finally, the product ViQu will be able to conduct interviews/viva(s), mark them, and a fine database of similar questions will be created for the above.

3.3 Work Breakdown Structure

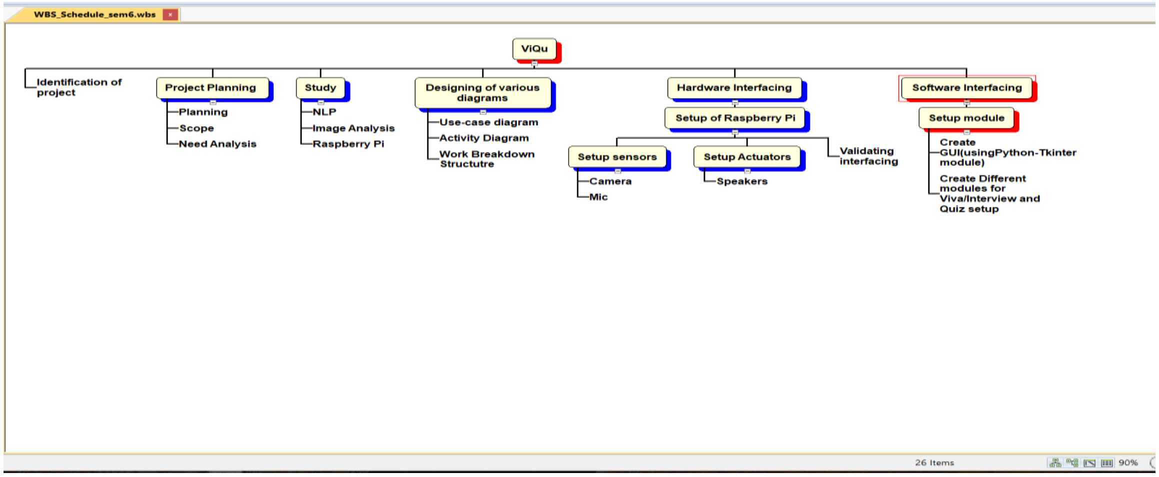


FIGURE 1: Work breakdown structure semester 6

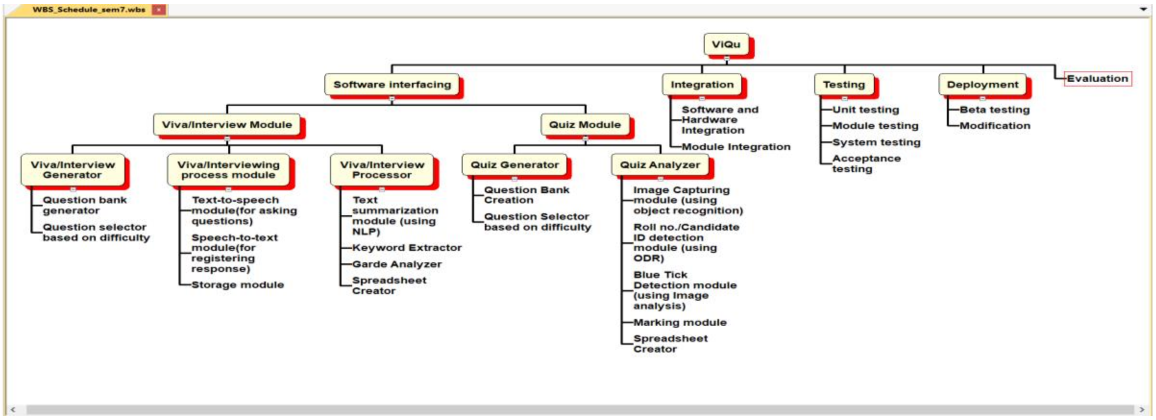


FIGURE 2: Work breakdown structure semester 7

3.4 Tools and Technologies Used

The various tools and technologies used in the project are as follows:

I. Raspberry Pi 3 model B

To make the product a stand-alone one the Raspberry Pi 3 model B will be used, single board computer which while being a simple and easy to use computer will also empower us to run the computationally heavy modules due to its quite high specifications.

The Raspberry Pi 3 Model B includes 802.11n WiFi, Bluetooth 4.0, and a quad-core 64-bit ARM Cortex A53 running at 1.2 GHz, RAM 1GB of LPDDR2-900 SDRAM, and the graphics capabilities, provided by the VideoCore IV GPU.

All of this will aid the processes working in the proposed system such as the text summarizer, online APIs for text-to-speech and speech-to-text, image analyzer, etc.

II. Hardware peripherals

Various peripherals used such as speakers, microphone, MicroSD card(16 GB), data cable and ethernet cable

III. Raspbian

It is a Debian based linux distro. An open source operating system for Raspberry Pi 3 model B which satisfies all the ned s of our project

IV. Speech Synthesis

These engines on the systems rely on the Google API and also the on-board espeak speech engine which are available using the python libraries. The library used here is “gTTS” which is a google text-to-speech library and “talkey” which is a wrapper to the espeak speech engine for linux distros. The text to be converted to speech is sent to the engine and a corresponding audio file is received.

Google has a great Speech Recognition API. This API converts spoken text (microphone) into written text (Python strings), briefly Speech to Text. You can simply speak in a microphone and Google API will translate this into written text. The API has excellent results for English language.

There are several APIs available to convert text-to-speech in python. One of such APIs is the Google Text to Speech API commonly known as the gTTS API. gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as a mp3 file.

The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow.

V. Python

All the basic coding is done in Python 2.7 owing to its ease of coding and availability of large number of libraries and support community.

VI. Python Libraries for services

Various python libraries are used to perform tasks such as speech recognition, speech synthesis, pattern maching etc.

VII. Speech Recognition

The library used here is “speech_recognition 3.8.1”. contains various speech recognition engines which were tested for accuracy and the one provided by Google Inc (free version)/Google Cloud Services (paid version), was found to be the most accurate amongst all of them. The microphone connected to the device is being used as the audio source which is used as a recording medium. This audio is then sent to the speech recognition engine and a text is received from the API.

VIII. Keyword matching

The libraries used here are “fuzzywuzzy”, “nltk” and “re”. Keyword matching is done via basic n-gram model which just does a fuzzy string comparison. The nltk and re libraries is used to create n-grams of the textual answer. The fuzzywuzzy library has a function fuzz.ratio() which gives us the matching ratio between two strings i.e. an approximate percentage match between two strings is obtained.

IX. Python libraries for hardware control

“time” is used to create required delays in the system process. “os” is used to perform operating system related tasks. The python standard library “shutil” works as a helper in various tasks. The “csv” library is used to handle .csv files.

DESIGN SPECIFICATIONS

4.1System Architecture

I. Use Case Template

TABLE 5: Use Case Template

Use Case ID:	UID-101		
Use Case Name:	Interviewing/Viva and Quiz system		
Created By:	CPG28	Last Updated By:	CPG28
Date Created:	April 2nd, 2018	Date Last Updated:	April 3rd, 2018

Primary Actor:	Examiner
Secondary Actor:	Student, Teacher,ViQu
Description:	Teacher needs to submit questions and answers to ViQu, ViQu will make the database of these Q/A.ViQu will use the database for quizzing and taking interview/viva, Students will take the interview and quiz and ViQu will evaluate them for interview/viva it will use text summarization (using natural language processing) and for quiz it will use image processing to find out marked answers, these answers will be submitted to teacher.
Preconditions:	1. Hardware should be working correctly. 2. Interview/Viva should be in English. 3. Proper image should be provided for checking the quizzes. 4. Q/A should be provided by the teacher. 5. Extra disk space may be required. 6. Presence of the examiner may be required.
Postconditions:	1. ViQu can calculate the marks according to the students' performance. 2. Teacher can view the marks allotted by the ViQu. 3. Teacher can update the marksheet, if needed. 4. Question bank can also be generated using the database.
Minimal Condition:	1. Database will be generated. 2. Quizzes will be generated. 3. Results will be provided.

Frequency of Use:	1. The process of Interviewing/Viva and Quiz depends upon the number of participants. 2. Question/Answer and updating of database will occur every month.
Trigger:	1. Providing Q/A will trigger Database. 2. Database will trigger Quiz making. 3. Taking interview will trigger taking interview/viva by ViQu. 4. Providing quiz pictures will trigger checking quizzes. 5. Any kind of evaluation will result in result generation.
Variations:	1. Question/Answer can be provided as text file or as excel file. 2. Pictures may be directly provided to the system instead of making ViQu take them.
Extensions:	End session if input not provided for too long.
Assumptions:	1. Students responds in English. 2. Pictures provided for checking viva should be in good quality.

II. System Architecture

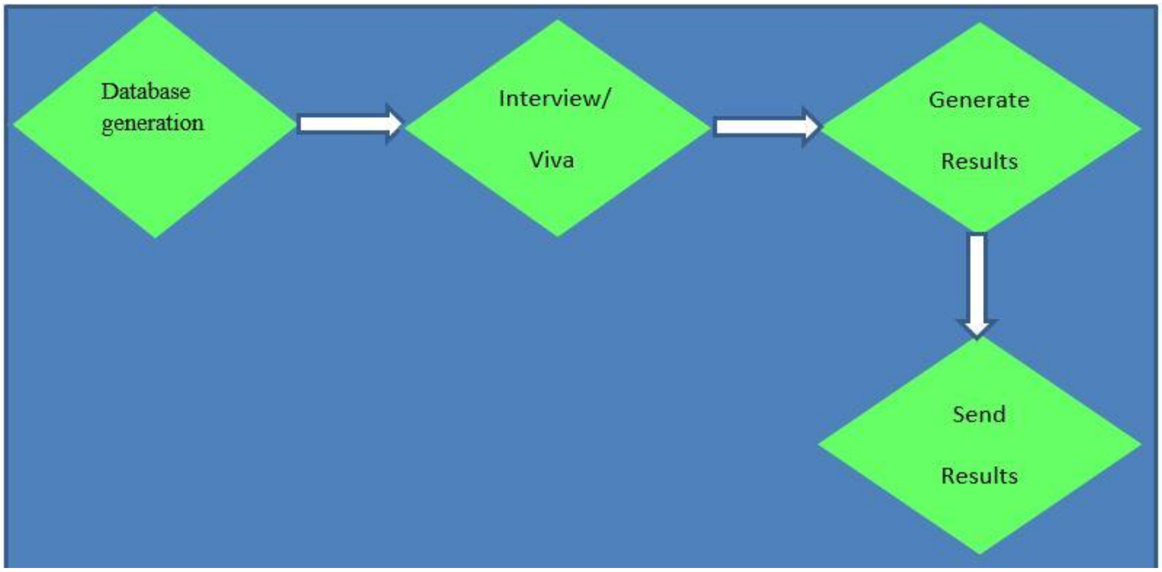


FIGURE 3: System Architecture

III. Context Diagram

It provides the bird-eye view of the project explaining what all is happening in the system particularly related to the data flow in the system.

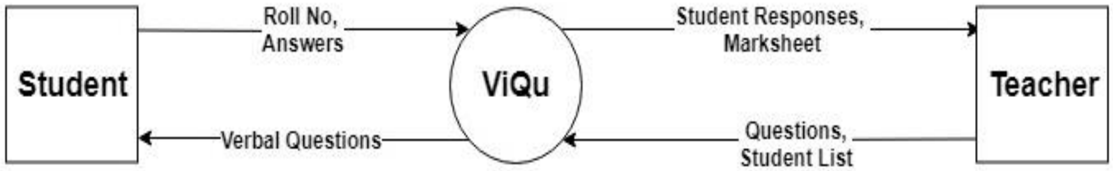


FIGURE 4: Context Diagram

IV. MVC Architecture: Model-View-Controller Architecture

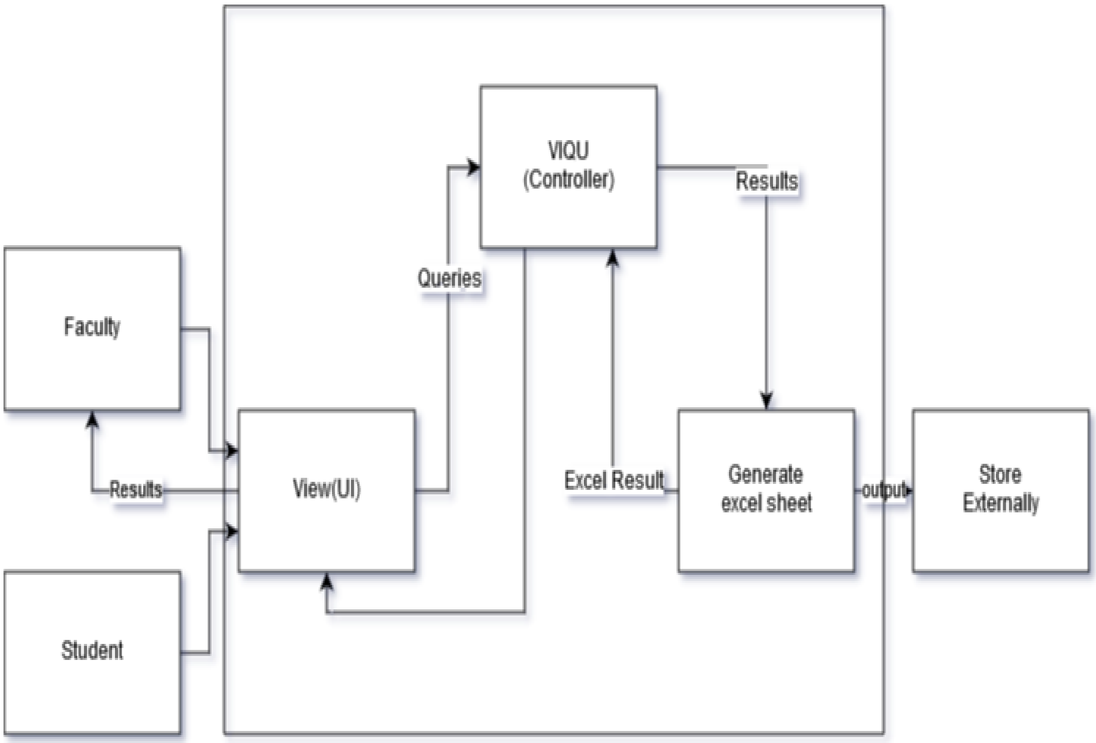


FIGURE 5: MVC Architecture

4.2 Design Level Diagrams

I. Activity Diagram

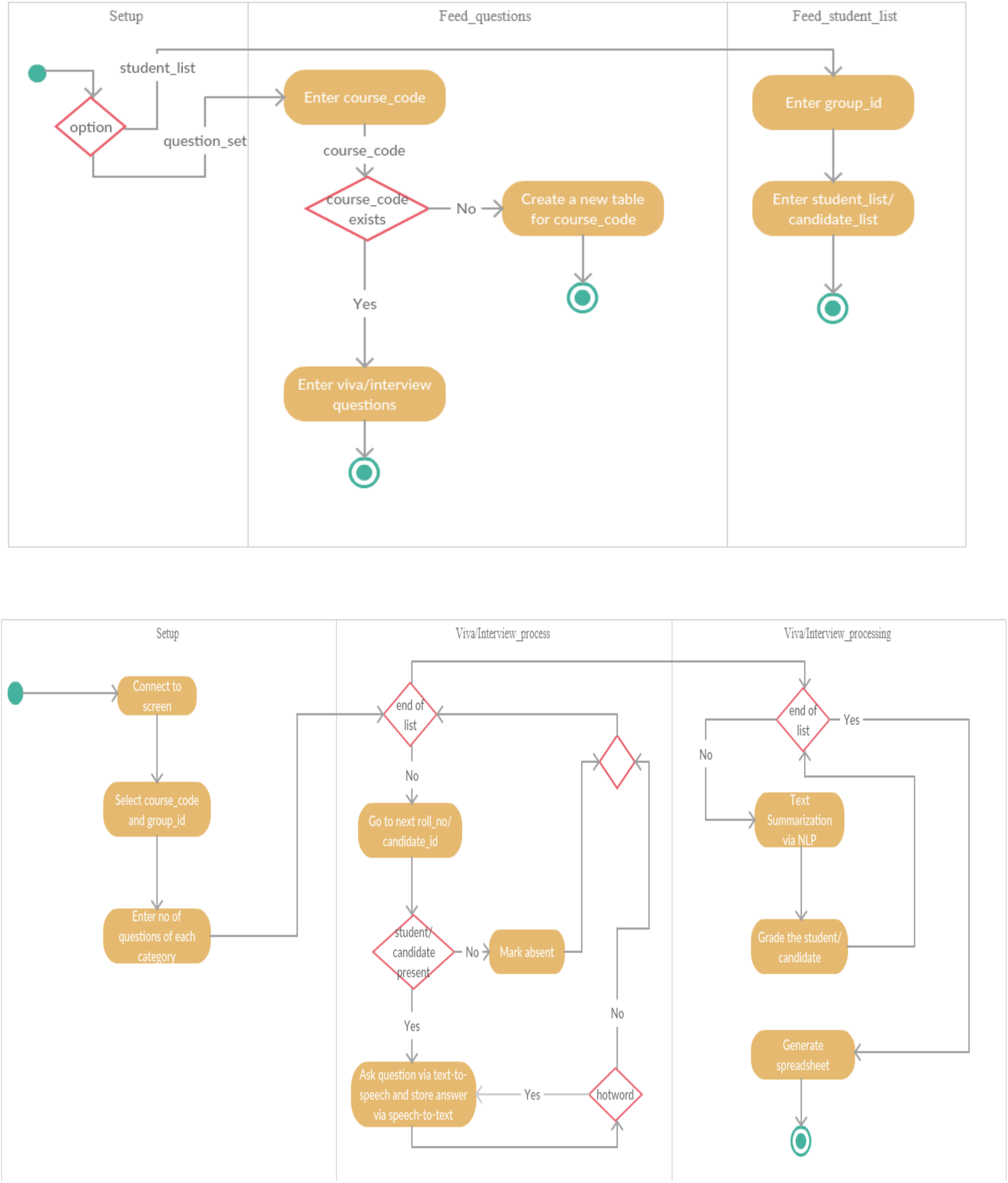

Setup
Feed_questions
Feed_student_list
student_list
option
question_set
Enter course_code
course_code
course_code exists
No
Create a new table for course_code
Yes
Enter viva/interview questions
Enter group_id
Enter student_list/ candidate_list
Setup
Viva/Interview_process
Viva/Interview_processing
Connect to screen
Select course_code and group_id
Enter no of questions of each category
end of list
No
Go to next roll_no/ candidate_id
student/ candidate present
No
Mark absent
Yes
Ask question via text-to-speech and store answer via speech-to-text
Yes
hotword
No
end of list
Yes
No
Text Summarization via NLP
Grade the student/ candidate
Generate spreadsheet

FIGURE 6: Activity Diagrams

II. Class Diagram

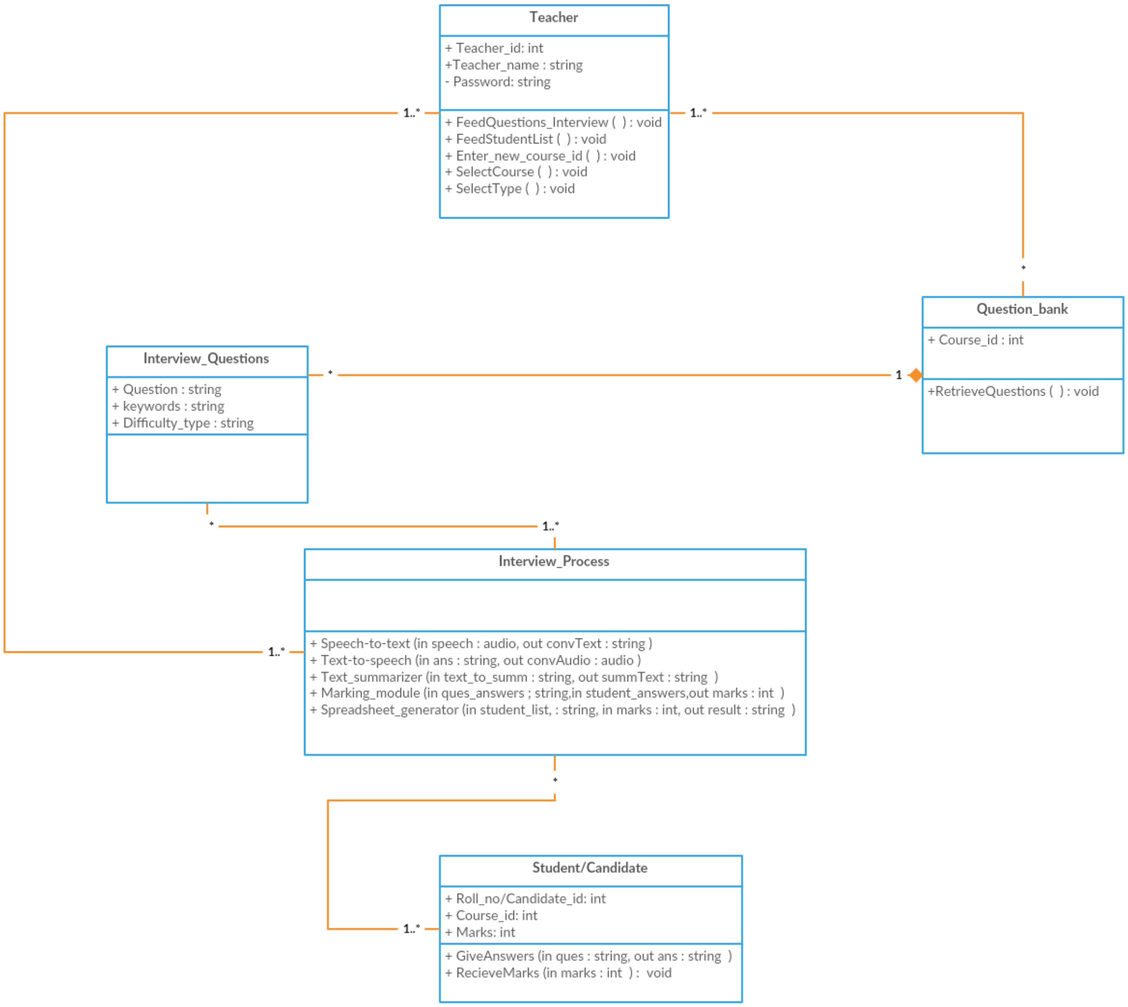

Teacher
+ Teacher_id: int
+Teacher_name : string
- Password: string
+ FeedQuestions_Interview ( ) : void
+ FeedStudentList ( ) : void
+ Enter_new_course_id ( ) : void
+ SelectCourse ( ) : void
+ SelectType ( ) : void
1..*
1..*
*
Question_bank
+ Course_id : int
+RetrieveQuestions ( ) : void
1
*
Interview_Questions
+ Question : string
+ keywords : string
+ Difficulty_type : string
*
1..*
Interview_Process
+ Speech-to-text (in speech : audio, out convText : string)
+ Text-to-speech (in ans : string, out convAudio : audio)
+ Text_summarizer (in text_to_summ : string, out summText : string)
+ Marking_module (in ques_answers ; string,in student_answers,out marks : int)
+ Spreadsheet_generator (in student_list, : string, in marks : int, out result : string)
1..*
*
Student/Candidate
+ Roll_no/Candidate_id: int
+ Course_id: int
+ Marks: int
+ GiveAnswers (in ques : string, out ans : string)
+ RecieveMarks (in marks : int) : void
1..*

FIGURE 7: Class Diagram

III. Data Flow Diagrams

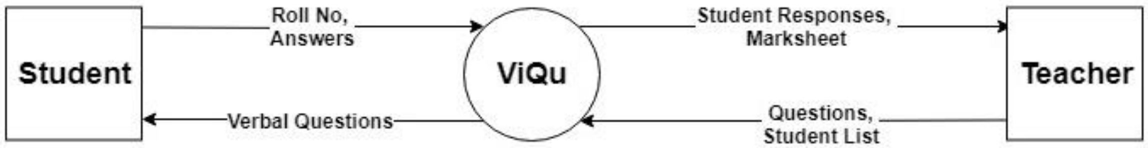


FIGURE 8: DFD level-0

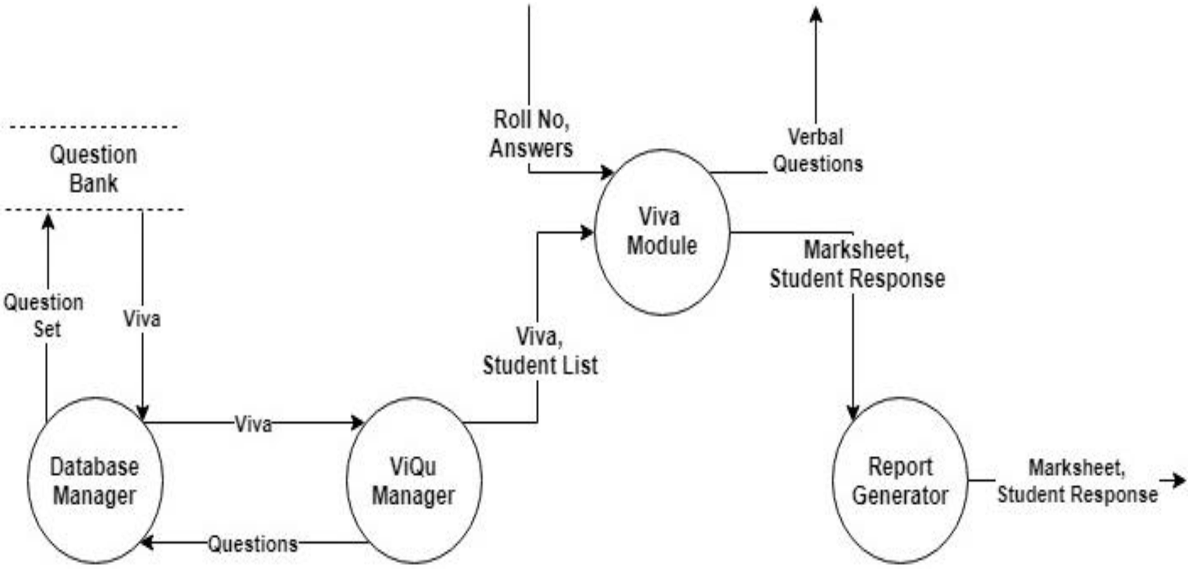


FIGURE 9: DFD level-1

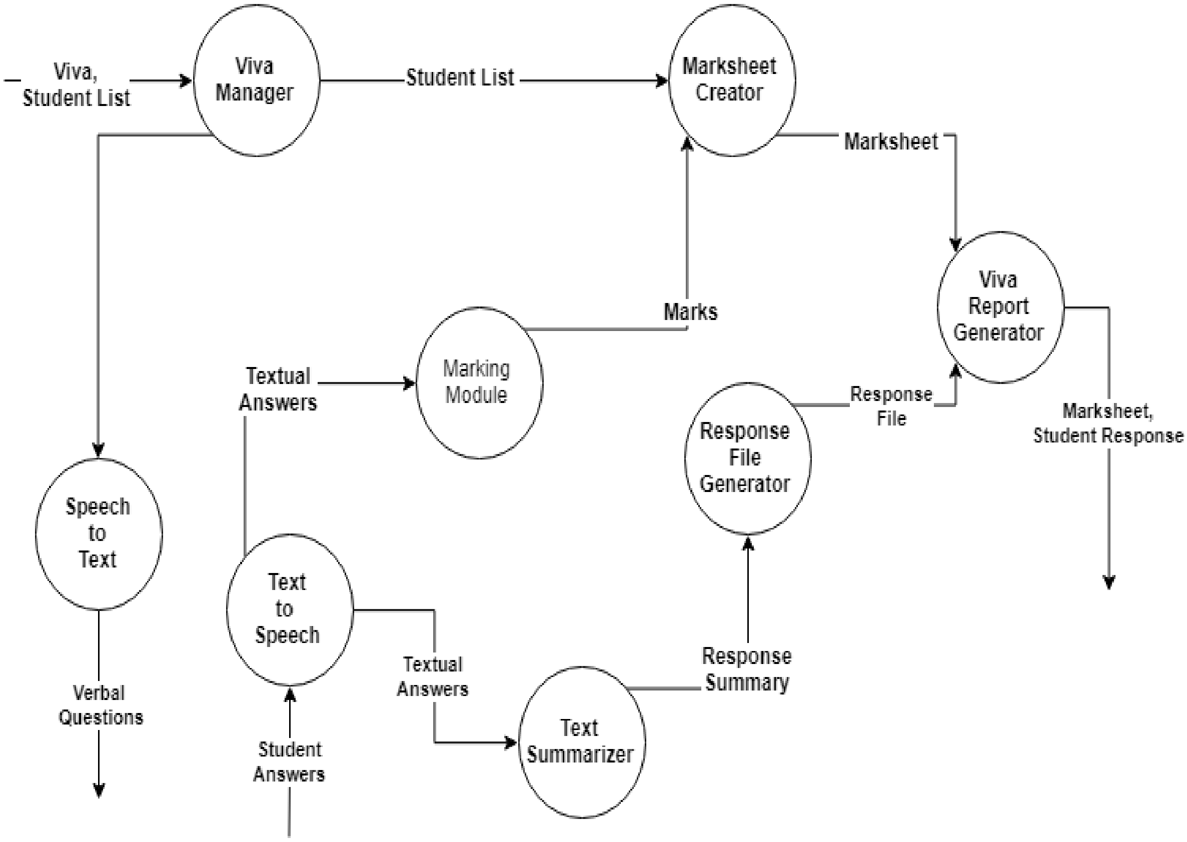


FIGURE 10: DFD level-2

4.3 User Interface Diagrams

I. Use case diagram

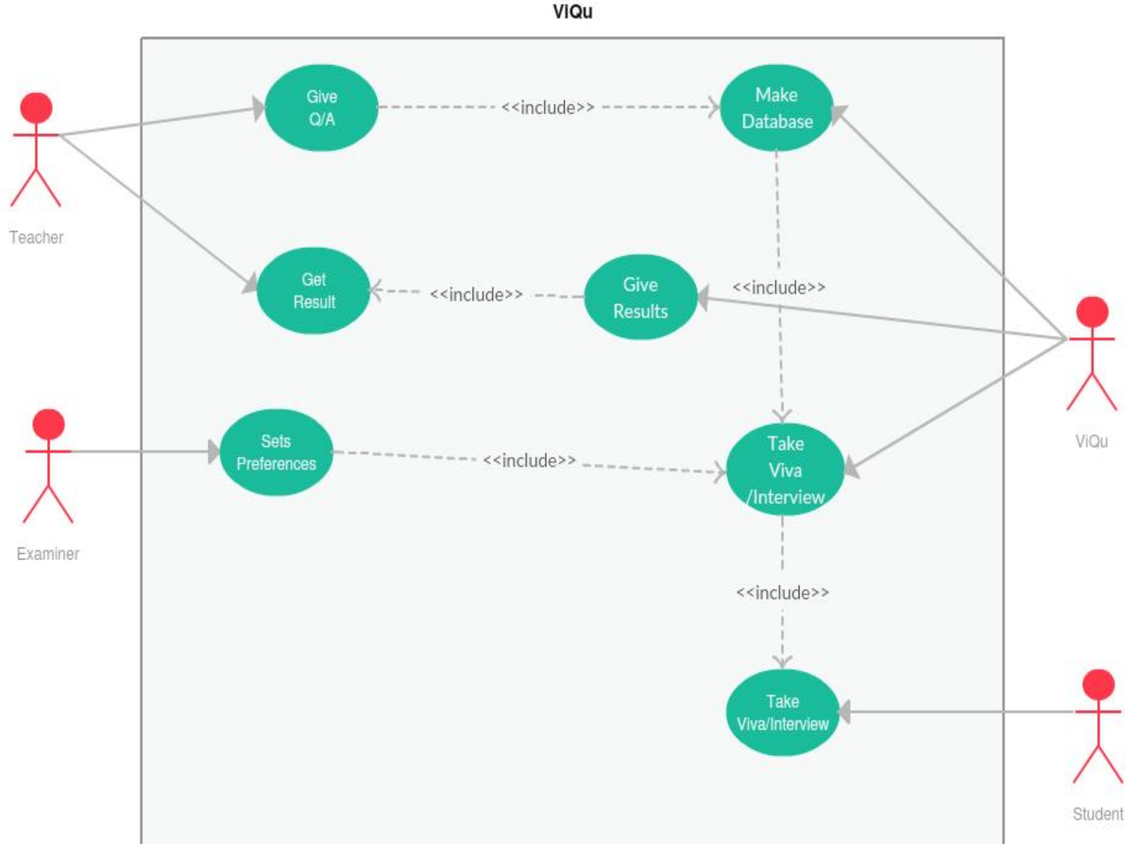


FIGURE 11: Use case diagram

4.4 System Screenshots

FIGURE 12: Peripherals Screenshot

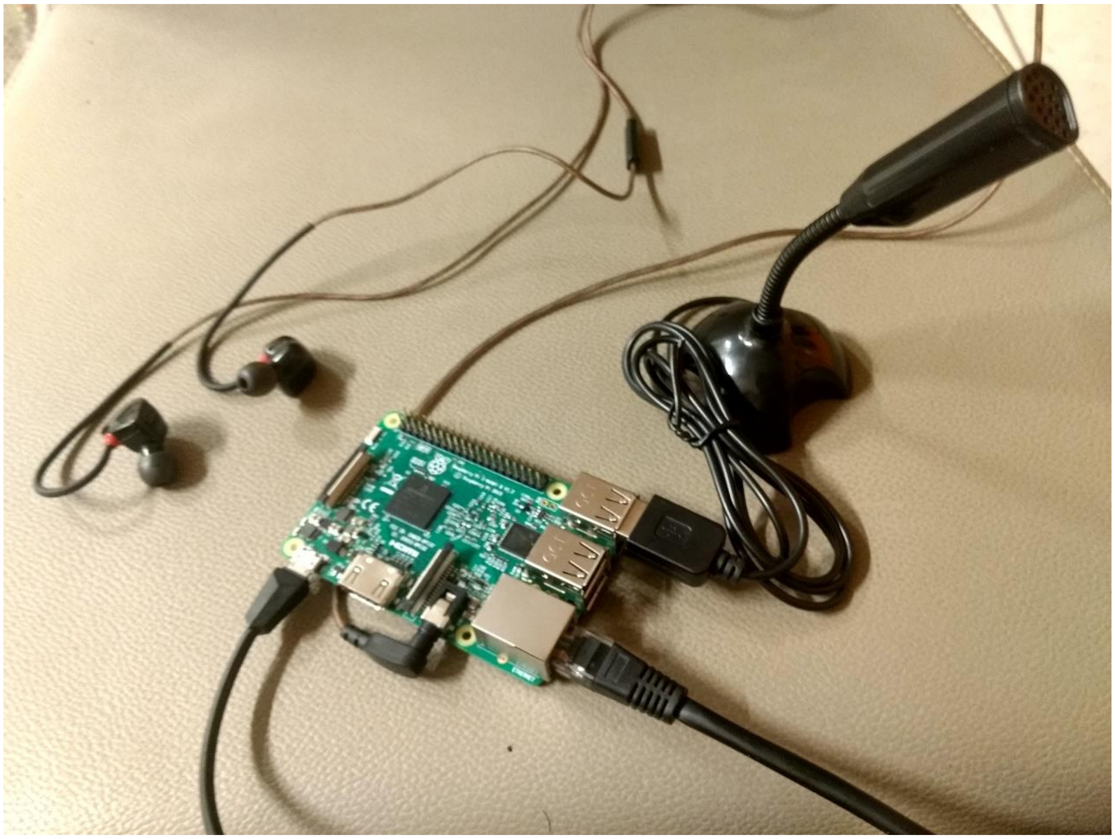
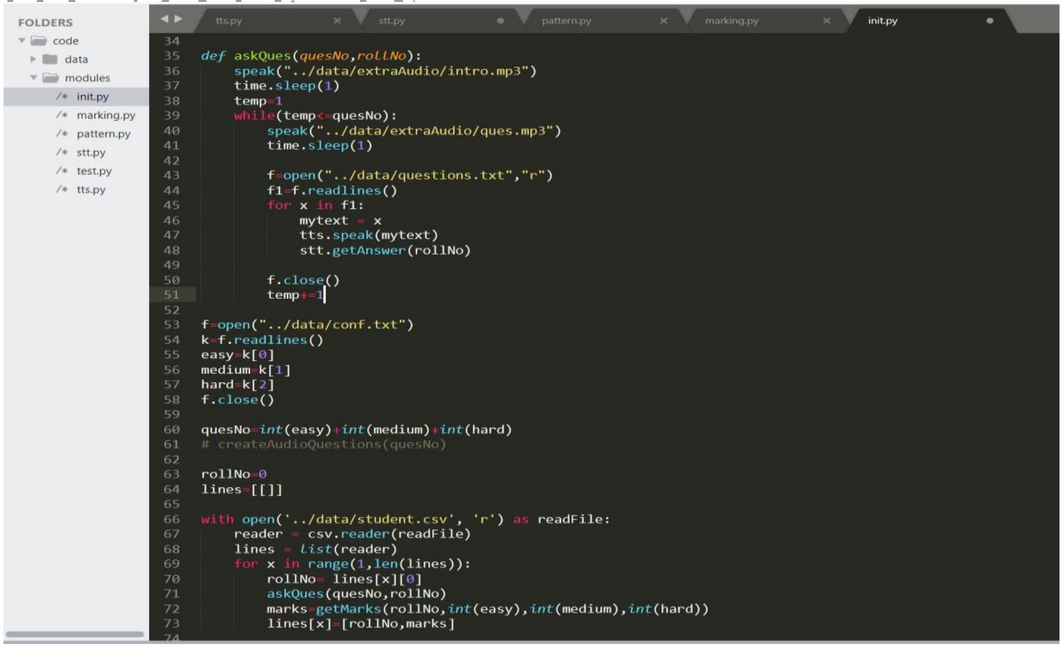


FIGURE 13: Raspberry Pi 3 screenshot



```

def askQues(quesNo,rollNo):
    speak("../data/extraAudio/intro.mp3")
    time.sleep(1)
    temp=1
    while(temp<=quesNo):
        speak("../data/extraAudio/ques.mp3")
        time.sleep(1)

        f=open("../data/questions.txt","r")
        f1=f.readlines()
        for x in f1:
            mytext = x
            tts.speak(mytext)
            stt.getAnswer(rollNo)

        f.close()
        temp+=1

f=open("../data/conf.txt")
k=f.readlines()
easy=k[0]
medium=k[1]
hard=k[2]
f.close()

quesNo=int(easy)+int(medium)+int(hard)
# createAudioQuestions(quesNo)

rollNo=0
lines=[[]]

with open('../data/student.csv', 'r') as readFile:
    reader = csv.reader(readFile)
    lines = list(reader)
    for x in range(1,len(lines)):
        rollNo= lines[x][0]
        askQues(quesNo,rollNo)
        marks=getMarks(rollNo,int(easy),int(medium),int(hard))
        lines[x]=[rollNo,marks]
```

FIGURE 14: Code Screenshot

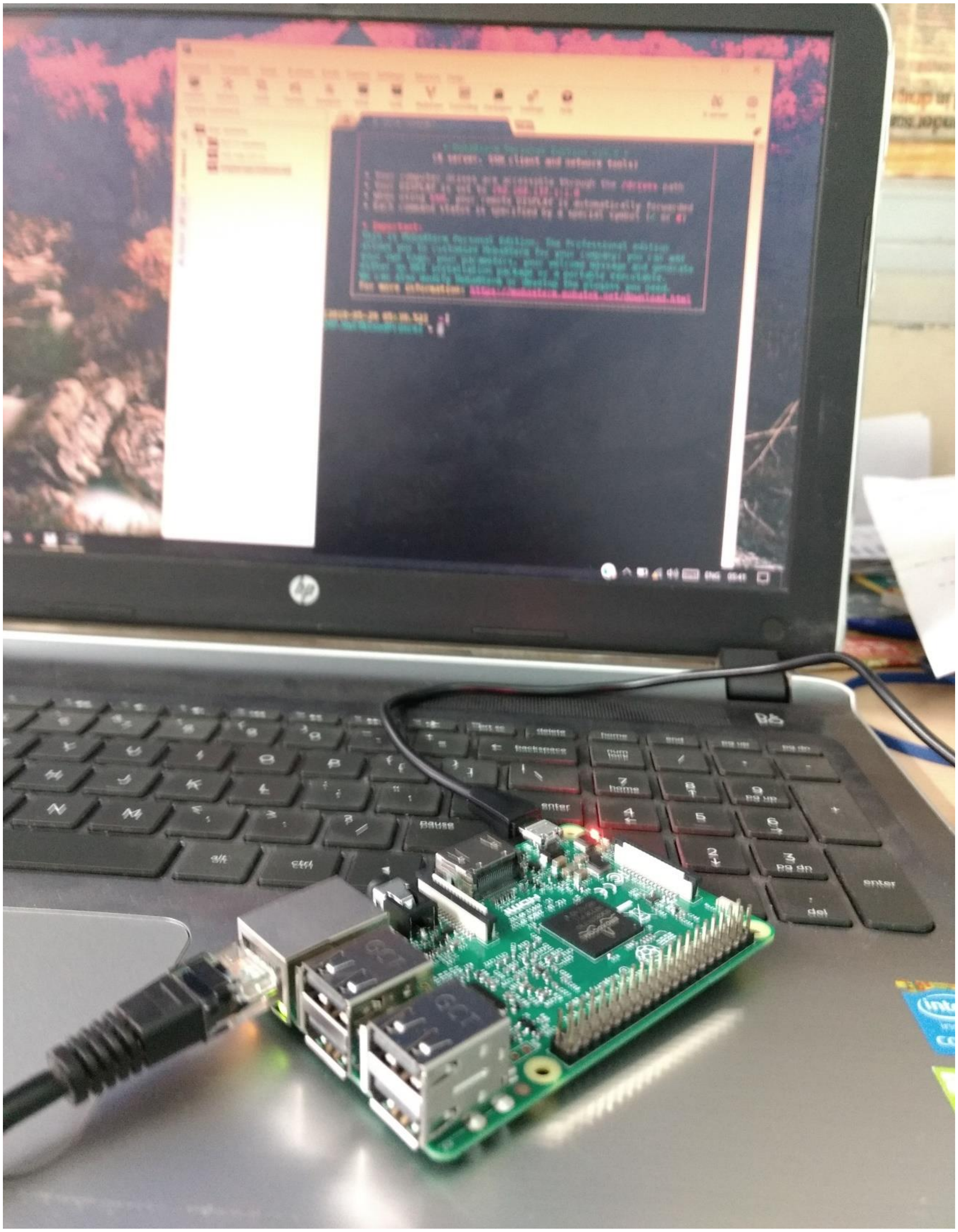


FIGURE 15: Raspberry Pi 3 connected with laptop screenshot

IMPLEMENTATION AND EXPERIMENTAL RESULTS

5.1 Experimental Setup

The experiment of the bot was run in phases with an increasing complexity. Firstly, unit testing was done in which the smallest testable parts of an application, called units, were individually and independently scrutinized for proper operation. The setup was to create the simpler parts of test cases to test the individual units for correctness. Some test cases simulating the conditions of the interview/viva were created at different levels in the software which were run rigorously on the software units. The smallest parts of software tested for correctness and then connected in an orderly fashion into bigger units which were again tested. Then the complete code was assembled together to create the software and tested on the whole for correctness. After this the parts of hardware were tested individually first and whether the different software modules being created are compatible with the available hardware or not. Appropriate amendments in the software system were made to accommodate the hardware specifications. Then the final assembly of the hardware was tested with the complete software.

5.2 Experimental Analysis

The experimental analysis is an important part of the project which need to be done with utmost precision. The analysis brings out the flaws in the system which are measured on the basis of various performance parameters depending on various internal and external factors of the system.

5.2.1 Data

Input data used for the testing process is the list of students for taking the interview/viva and a set of questions and answers against them which need to be asked in the testing phase with all the possible types of questions possible in the system. As for the output the results of the students i.e. the marks of the students against their roll numbers are stored in a spreadsheet and along with it the transcribed answers of the students are stored in individual text files which acts as result to the invigilators.

5.2.2 Performance parameters

The prime role of this bot is to conduct interview/viva of candidates which is the most important aspect to be checked in the testing phase. As for the quantitative metrics there are various metrics of the hardware and the software itself.

I. The correctness of the transcribed answers of the candidates which is dependent on the accuracy and correctness of google cloud-based speech engine which is used via the speech_recognition python wrapper. As per the documentation of the speech engine it has close to 9% error rate which is lowest in the market.
II. The marking module depends on the string pattern matching module which needs to incorporate word variations such as plurals. A certain threshold needs to be set in the system to partially match the words with the provided answers.
III. The different hardware systems have certain performance parameters of their own which need to be met such as the speaker must have a clear output, the microphone must be able to detect normal human voice and moreover the complete system must run the software in an acceptable level of performance speed wise.

5.3 Testing Process

Testing process helps in realizing the flaws in the system and is an important part of the project creation.

5.3.1 Test Plan

The testing was done as explained below.

5.3.1.1 Features to be tested

The following features were identified to be tested:

I. Text-to-Speech

To test for the correctness of speech synthesis modules of the software along with their conjunction with the relevant hardware components i.e. speaker.

II. Speech-to-Text

To test for the correctness of speech recognition modules of the software along with their conjunction with the relevant hardware components i.e. microphone.

III. Marking module

To test the marking module to generate marks of the students accordingly and test the string pattern matching module for correctness and set the appropriate threshold for partial matching.

IV. Result creation

To check if the spreadsheet is being created correctly and completely.

V. Creation of individual result files

To test for creation of individual files with the answers of each candidate in digital format to the invigilator.

5.3.1.2 Test Strategy

Firstly, unit testing was done in which the smallest testable parts of an application, called units, were individually and independently scrutinized for proper operation. The smallest parts of software tested for correctness and then connected in an orderly fashion into bigger units which were again tested. Then the complete code was assembled together to create the software and tested on the whole for correctness. After this the parts of hardware were tested individually first and whether the different software modules being created are compatible with the available hardware or not. Appropriate amendments in the software system were made to accommodate the hardware specifications. Then the final assembly of the hardware was tested with the complete software.

5.3.1.3 Test Techniques

I. Create test cases and test data for the unit testing of software components. These test cases have the work flow of the relevant components and test these units for testing.

II. Then the individual units are combined to form larger code segments and test them again under the input set required and appropriate results are looked for. Further the complete software is compiled to completion and tested with complete use cases.

III. The hardware components are then tested individuality for checking the devices along with testing the individual components with the help of python codes so as to check the compatibility of various hardware components with the software modules.

IV. Finally, the complete software is tested upon the hardware for the final testing. To do this a sample interview/viva was created which was tested on the final product created and run to check for correctness. A sample student data sheet, questionnaire is created and an output was obtained which was tested for the correctness.

5.3.2 Test Cases

There are some test cases created according to the different conditions of input data. Conditions being different number of candidates and different type and number of questions in the interview/viva. Following are the test cases created:

I. Single candidate and single question
II. Single candidate and multiple questions
III. Single candidate and multiple questions of different types
IV. Multiple candidates and single question
V. Multiple candidate and multiple questions
VI. Multiple candidate and multiple questions of different types

5.3.3 Test Results

The testing phase brought out some of the shortcomings of the systems such as the python libraries which were inconsistent with the hardware, hardware shortcomings of speed and accuracy, etc. Certain problems with the flow of the process were also highlighted. All these problems were corrected after the test and tested again. However, the final results were found to be totally correct.

5.4 Results and discussions

The bot is made up of Raspberry Pi 3 model B which firstly needed to be calibrated properly according to the needs where modules such as speaker and microphone were to be adjusted accordingly. Most importantly certain modules which worked fine on the larger computer system did not work well on the small embedded system created. So, in such modules alternate solutions were found out for example in some cases espeak speech engine was used instead of google cloud-based speech engine.

The problem of partial match of the keywords was also realized during the unit testing of the pattern matching module and which was eliminated, as mentioned above, by using the fuzzywuzzy technique to detect partial match of the string and then setting a threshold to detect the word variations.

During the testing a very crucial drawback that is the software required quite a lot of time to load into the system and start the whole process. This was due to on-board speech engine "espeak". But no other alternative seemed to work on the mini embedded system. However, the system parts worked perfectly in terms of correctness and generated the result sheet as was required. Finally, it was found that the system worked perfectly in conjunction to give the desired outcome.

5.5 Inferences Drawn

The bot which has been created works perfectly to give the desired output. It works well on the single word answers, fill ups, true/false and multiple-choice questions. And transcribes the answers well with an acceptable error in speech recognition and marks the answers accordingly with an acceptable amount of error.

It must be mentioned that a lot of future work goes into creating a more advanced bot which may take interviews/viva(s) where single or multiple line answers may also be received which requires some semantic analysis of the written text to be done for the marking module. Moreover, in further work more subjects may also be included in the system.

5.6 Validation of Objectives

TABLE 6: Validation of objectives

S. No.	Objectivess	Status
1	Create a hardware-based bot which can conduct an interview/viva	Successful
2	To ask questions via TTS: text-to-speech	Successful
3	To receive answers via STT: speech-to-text	Successful
4	To mark the answers based upon the keywords to the questions provided	Successful
5	To create a final marksheet with student roll number and marks	Successful
6	To provide the answers of each candidate in digital format to the invigilator	Successful
7	Eventually create a question bank of relevant questions	Unsuccessful
8	To improve our skill set	Successful

CONCLUSIONS AND FUTURE DIRECTIONS

6.1 Conclusions

The overall interview/viva process was quite a tedious and tiresome task which needed to be made easier and simpler with a relatively cheaper solution. So, finally a raspberry pi 3 based hardware bot has been created to solve the above problem. The complete process of interviewing has been made easier and has been quite automated and the overall task of marking has also been simplified as the marksheet of the complete set of candidates is created all at once, reducing the time in results declaration. Moreover, the answers of the candidates are digitally stored for further references.

6.2 Environmental, Economic and Societal benefits

I. Environmental Benefits

ViQu saves the use of paper to ask the questions which is traditionally done using the paper, ViQu asks the voice question reducing the need of paper. ViQu can work perfectly on 5V and 2.5A, which saves a lot of electricity. ViQu can take interviews/viva(s) anytime and anywhere which also saves the time and cost to travel, and also a human may not be available at sometimes. ViQu is a hand-held device, so no additional transportation is required.

II. Economic Benefits

ViQu can take interviews without any charge for subscriptions, compared to other competitors who charge hundreds of dollars for a month. ViQu is a hand-held device, so there is no additional cost required for its transportation. It is an independent system which saves us the cost for installing the dependencies. ViQu can work perfectly on 5V and 2.5A, which saves a lot of electricity and thus save money.

III. Societal Benefits

ViQu can maintain the accuracy of taking the interview of all the candidates in a proper unbiased manner compared to human who may have deep rooted biasness when it comes to analysis. ViQu is available to take interviews anytime and anywhere which a human might not be able to. ViQu can evaluate the candidate in real time and giving the results just after the interviews are completed.

6.3 Reflections

Projects like this are very crucial in a student's life as they widen our domain of practical knowledge, help us to use our theoretical knowledge more efficiently and learn new concepts. This project has helped us to increase our skill set. Choosing a project was a highly brainstorming activity. Having healthy discussions among the team members after lot of research made our knowledge domain a little wider. Our learnings were successfully applied from Natural Language Processing to successfully perform speech-to-text and text-to-speech conversions. Our basic concepts of programming were applied to create a pattern matcher and a marking module. Setting up Raspberry Pi3 was altogether a very new, interesting and informative experience. Running text-to-speech module on Raspberry Pi3 introduced us to the basic working of the Single board computer. Modelling our project using diagrams made us realize the various aspects of our project. The team learned how to make a good and proper technical report.

6.4 Future Work

ViQu has a great future scope due to the vast application areas of the individual techniques used for making it, and some of those applications are mentioned below:

I. Interview Bot

Interviews are not just about the domain knowledge but are also about our body language and if the candidate can work properly under pressure and many more things, these all things can be done using ViQu by adding a camera and a deep learning model to detect the body language.

II. Non-Technical Interviews

Not just Technical Interviews, by applying Machine Learning, deep learning and semantic analysis non-technical interviews may be taken as well, eliminating the need of a human being in the entire process of campus recruitment.

III. Interview/viva of non-objective questions

By using semantic analysis, interview/viva comprising of not only objective questions can be taken (fill in the blanks, true/false, multiple choice questions, one-word answers) but also subjective questions.

IV. Stress Management

Going beyond the technical field our bot may be used in the field of psychology. By applying machine learning and semantic analysis, stress level of people could be calculated and if found above a particular level, give them tips on reducing their stress.

7.1 Challenges Faced

I. Hardware related

Configuring the raspberry pi: Configuring the raspberry Pi was the biggest challenge as the team was handling this hardware for the first time, team must rely on other sources like blogs and YouTube videos to configure the raspberry according to their needs. For example: connecting raspberry pi with PC using HDMI, connecting raspberry pi with PC using ethernet cables and how to use the wifi card of the raspberry pi.

Getting the proper microphone: Our project required the use microphone and was ordered, but this microphone did not perform up to the mark and was also sometimes working and sometimes not, so the microphone must be replaced to get a new one which finally did the job.

II. Software related

Selecting OS for raspberry pi: There are different operating systems available for raspberry pi, and some operating system support the required hardware and some not, finally Raspbian OS was used which was fine with the system. For example: on noobs operating system the microphone was not working properly.

Installing the python libraries on raspberry pi: There were many dependencies which were not resolved for some of the libraries and were very difficult to resolve as either the solutions provided were of a very older version or they didn’t worked on raspberry pi.For example: for speech_recognition library PyAudio 0.2.11 was the dependency and the raspberry pi contained PyAudio 0.2.9 and was not upgradable.

Interfacing the different modules on raspberry pi: As all the codes were made on the laptop when it was transferred to the raspberry pi it was difficult to run the same codes on raspberry pi and also as the laptops are pretty fast in executing the task but raspberry pi is not that fast due to which there was significant delays while executing codes on raspberry pi.

III. Team related

There were no such major challenges related to the team as the communication between the team members were good and also the workload was divided properly on all of us and our work was done before the deadline.

IV. Project related

Creating report: This was a very difficult part of the project were required to make the report according to the sample template provided and this sample template was very specific about the formatting style and referencing, and this was the first time the team was making this kind of reports.

V. Product related

Unavailability of a proper microphone: Finding a microphone which was compatible with the raspberry pi was difficult to find and also there were some audio drivers missing in the raspberry pi required for the microphone.

Volume output of the raspberry pi: The volume output through the raspberry pi was low, as the power required by the audio device is significantly high than the raspberry pi can provide.

7.2 Relevant Subjects

The subjects that were found relevant or were helpful in creating our project are mentioned below:

Table 7: Relevant subjects

Subject Code	Subject Name	Description
UCS614	Embedded Systems Design	It gave us an aptitude to understand the inner workings of the hardware and how the connections can be made
UML602	Natural Language Processing	It was of key importance to our software system as all text manipulations, speech-to-text, text-to-speech and string pattern matching were done with the prior knowledge of natural language processing.
UTA011	Engineering Design - III	It helped in understanding raspberry pi functioning as the subject taught us the basics of

		Arduino Uno which gave us an experience to work with mini chip-based computers.
UCS406	Data Structures and Algorithms	It subject aided in creating the basic code and making the marking module itself.
UML501	Machine Learning	This subject proved to be a crucial part of the project as all the coding was done in python due to the ease of coding in python and availability of community support and large number of libraries
UCS503	Software Engineering	This subject helped us to understand the importance of scheduling the work of software, how to plan our goals, way to achieve them, understanding the process of UML diagram creation and how to make a good documentation.

7.3 Interdisciplinary Knowledge Sharing

Software Engineering concepts to analyze and design our project. UML diagrams, activity diagrams and class diagrams were created for better understanding of the work needed to be done for the successful execution of this project. Embedded system and engineering design gave us an aptitude to understand the inner workings of the hardware and how the connections can be made. whereas all the concept was designed with the help of natural language processing with all the coding been done in python.

7.4 Peer Assessment Matrix

TABLE 8: Peer Assessment Matrix

		Evaluation of		
		Shivam Sharma	Shobhit Jain	Shreya Aggarwal
Evaluation By	Shivam Sharma	4	4	4
	Shobhit Jain	4	4	4
	Shreya Aggarwal	4	4	4

7.5 Role Playing and Work Schedule

1. Shivam Sharma: Contributed in hardware integration of Raspberry Pi along with all its peripherals, interview/viva module and unit and integration testing.

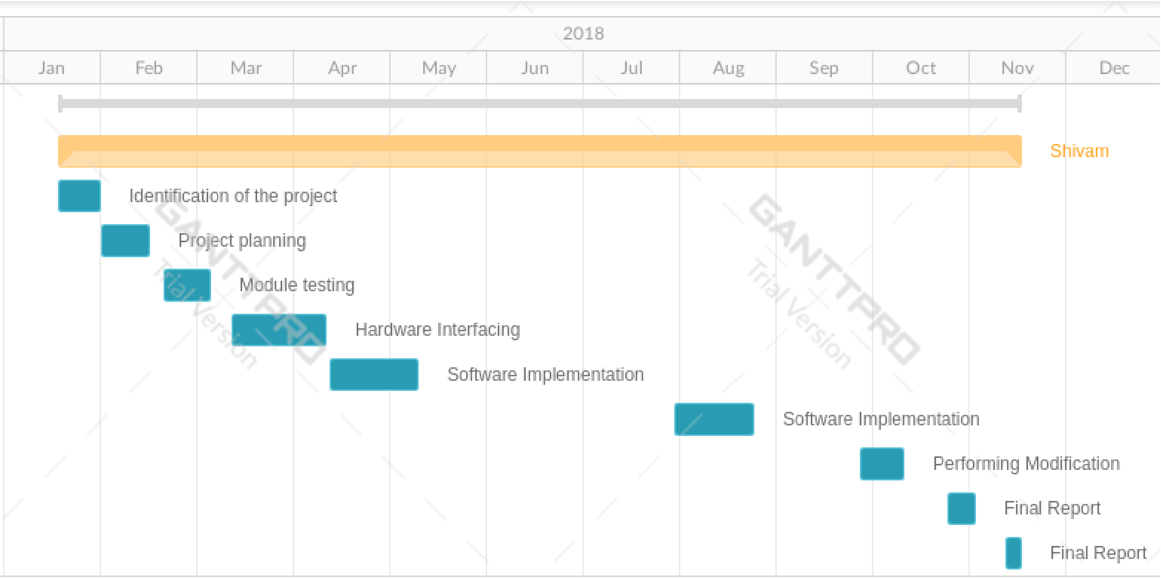


FIGURE 16: Shivam Sharma Gantt chart

2. Shobhit Jain: Contributed in hardware integration like the setup of Raspberry Pi, interview/viva module and software testing.

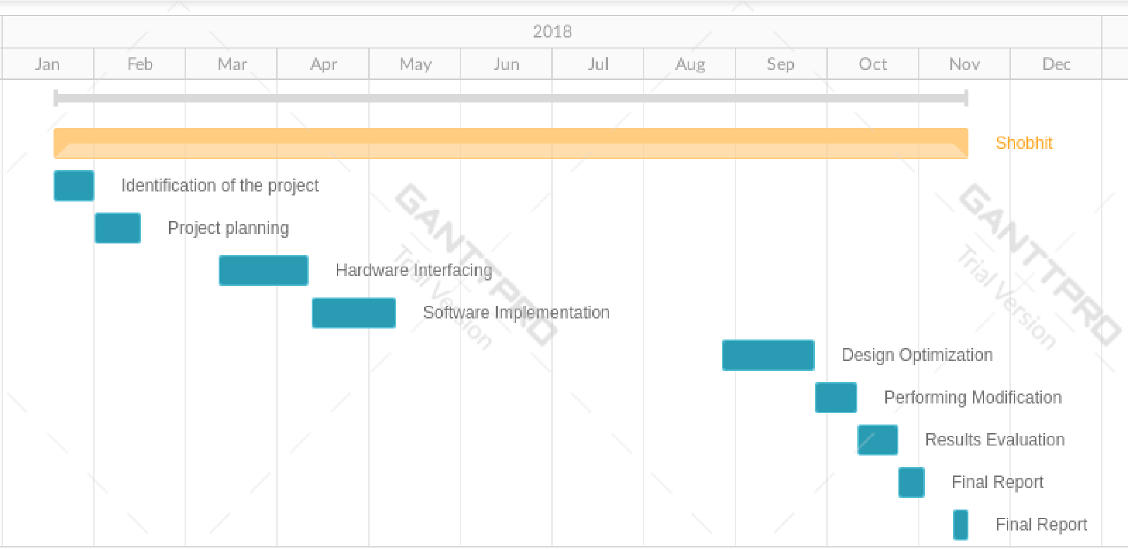


FIGURE 17: Shobhit Jain Gantt chart

3. Shreya Aggarwal: Contributed in interview/viva module, software testing and integration testing along with documentation.

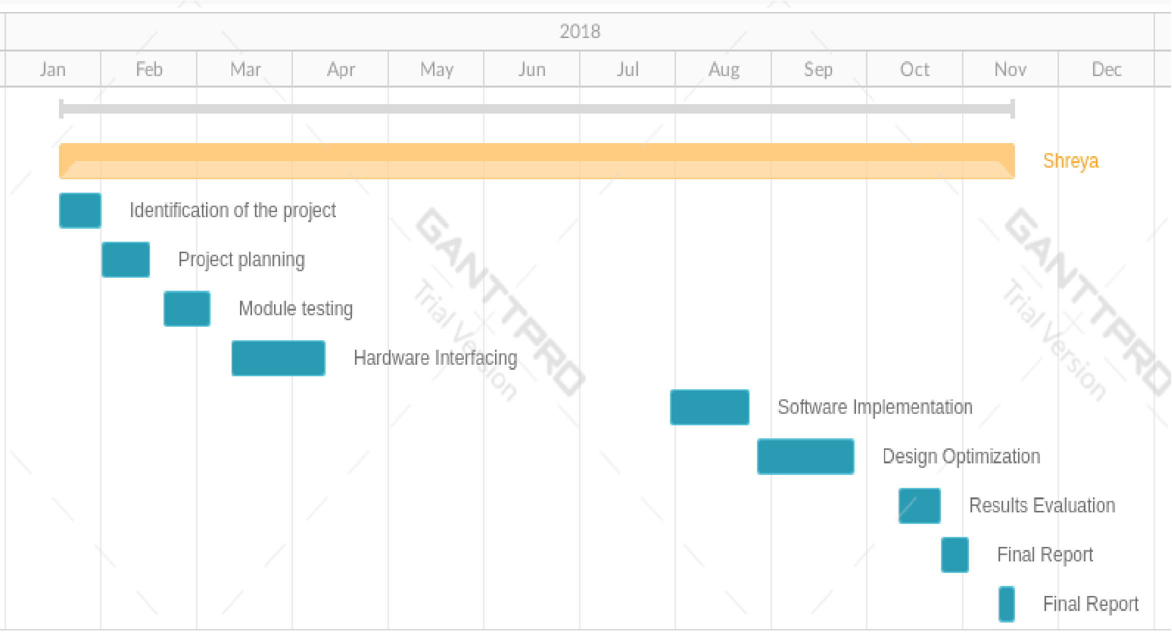


FIGURE 18: Shreya Aggarwal Gantt chart

TABLE 9: Individual Roles

Team Member Name	Hardware Integration	Interview/Viva Module	Software Testing	Integration Testing
Shivam Sharma 101503208	✓	✓		✓
Shobhit Jain 101503211	✓	✓	✓	
Shreya Aggarwal 101503213		✓	✓	✓

TABLE 10: Work Schedule

S. No	Activity	Month	January		February			March			April				May	
		Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Identification of project		■	■												
2	Planning of project and feasibility study (technical – software and hardware and, economic)				■	■										
3	Module testing						■	■								
4	Hardware interfacing								■	■	■	■				
5	Software implementation												■	■	■	■

S. No	Activity	Month	July	August				September			October				November	
		Week	15	16	17	18	19	20	21	22	23	24	25	26	27	28
5	Software implementation		■	■	■	■										
6	Design optimization						■	■	■							
7	Performing Modifications									■	■	■				
8	Results Evaluation												■	■		
9	Final Report														■	■

7.6 Student Outcomes description and Performance Indicators (A-K Mapping)

TABLE 11: A-K Mapping

SO	Description	Outcome
A1	Applying mathematical concepts to obtain analytical and numerical solutions.	Used mathematical concepts in calculating the fuzzy ratio and also in marking of answers and hence evaluating the final score of students.
A3	Applying engineering techniques for solving computing problems.	The project makes use of raspberry pi, text-to-speech, speech-to-text, n grams, fuzzywuzzy, python libraries and basic programming concepts to solve computing problems.
B1	Identify the constraints, assumptions and models for the problems.	For better efficiency, the environment during the time of interview/viva must be noise free, candidates should speak proper English and the teacher must follow the given file system.
B3	Analyze and interpret results with respect to assumptions, constraints and theory.	Considering the theory of project, assumptions and constraints identified, test cases were formulated and hence test results were noted.
C1	Design software system to address desired needs in different problem domains.	A hardware bot which conducts interview/viva of a given list of candidates according to given questions was created to solve different problems which were identified.
C2	Can understand scope and constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.	The interview/viva bot saves time, paper, electricity, money, reduces human intervention, and has easy maintenance.
D2	Can play different roles as a team player.	Each team member showcased different roles at different times.
E1	Identify engineering problems.	The tedious and redundant process of interview/viva was identified. This process also leads to delayed results.
E2	Develop appropriate models to formulate solutions.	A hardware bot which conducts interview/viva comprising of questions of categories – fill in the blanks, true/false, multiple choice questions and one-word answers is created.
E3	Use analytical and computational methods to obtain solutions.	The project uses analytical and computational methods like text-to-speech, speech-to-text, fuzzywuzzy, n grams and basic programming comcepts to obtain solutions.
F1	Showcase professional responsibility while interacting with peers and professional communities.	The team members showcased professional behavior while interacting with peers and professional communities.

G1	Produce a variety of documents such as laboratory or project reports using appropriate formats.	A technical report was prepared.
G2	Deliver well-organized and effective oral presentation.	A well-organized presentation was delivered.
H1	Aware of environmental and societal impact of engineering solutions.	Environmental benefits like saving of paper and electricity were noticed. The project has societal benefits like fast results and reduction of human intervention in interview/viva process.
H2	Examine economic tradeoffs in computing systems.	The hardware bot is cheaper and has easy maintenance. It saves power and hence money also.
I1	Able to explore and utilize resources to enhance self-learning.	Learnt various new concepts and techniques in different domains with the help of mentors and internet.
I2	Recognize the importance of life-long learning.	Learning new technologies and concepts made the team members realize learning is life-long and one should never stop learning.
K1	Write code in different programming languages.	Coding was done in Python 2.7.
K2	Apply different data structures and algorithmic techniques.	Linked list and Tuple (python) were used to store students' information, answers, questions, keywords and score.
K3	Use software tools necessary for computer engineering domain	MobaXTerm and Angry IP Scanner were used.

7.7 Brief Analytical Assessment

Q1. What sources of information did your team explore to arrive at the list of possible Project Problems?

Ans: For choosing a project, the team thought of various existing problems and their possible solutions. The team took help from the project mentors, teachers and internet to arrive at the list of possible project problems. Choosing a project was a highly brainstorming activity. After exploring many possibilities, the team chose to solve the problem of tedious, redundant and time-consuming process of interview/viva.

Q2. What analytical, computational and/or experimental methods did your project team use to obtain solutions to the problems in the project?

Ans: A hardware bot using Raspberry Pi 3 and Raspbian OS was created. For asking questions, text-to-speech conversion was applied and for recognizing answers speech-to-text conversion was done using gTTs and Speech Recognition of Google. To evaluate answers, pattern matching was done using n grams and fuzzywuzzy ratio. For marking the answers, basic programming skills were used. The coding is done in Python 2.7.

Q3. Did the project demand demonstration of knowledge of fundamentals, scientific and/or engineering principles? If yes, how did you apply?

Ans: The project required various fundamentals and engineering principles for its successful completion. After a lot of research and with the knowledge of basic Natural Language Processing principles gTTs for text-to-speech conversion (for asking questions), Speech Recognition of Google for speech-to-text conversion (for recognizing answers), n grams and fuzzy ratio for pattern matching were used. Also with the knowledge of Engineering Design and embedded systems, successful use of Raspberry pi 3 was made. Basic programming knowledge helped to code in python and mark answers for generating results.

Q4. How did your team shares responsibility and communicate the information of schedule with others in team to coordinate design and manufacturing dependencies?

Ans: Regular team meetings were held where the work accomplished was discussed along with the scheduling of future work. The team was regularly connected via mobile phone and e mail. Each team member would give his/her preference, then team coordinated and shared the project work among the team members. Special consideration was taken into place to ensure each team member has equal contribution.

Q5. What resources did you use to learn new materials not taught in class for the course of the project?

Ans: The team discussed with the project mentors, explored the World Wide Web through Google and took help from Thapar Institute of Engineering and Technology's library.

Q6. Does the project make you appreciate the need to solve problems in real life using engineering and could the project development make you proficient with software development tools and environments?

Ans: Projects like this are very crucial in a student's life as they widen our domain of practical knowledge, helps to use theoretical knowledge more efficiently and learn new concepts. Having healthy discussions among the team members after lot of research made everyone's knowledge domain wider. Learnings from various subjects like NLP, Engineering Design and Software Engineering were successfully applied to complete different modules of the project and made the team more familiar and proficient with software development tools and environments.