**ViQu – A smart Interview/viva and quizzing solution**

**Capstone Project Proposal**

**Submitted by:**

101503208 Shivam Sharma, BE Third Year, COE-10

101503211 Shobhit Jain, BE Third Year, COE-10

101503213 Shreya Aggarwal, BE Third Year, COE-10

Capstone Team No.: CPG28

Under the Mentorship of

Dr. Maninder Kaur & Mr. Ashish Girdhar

Assistant Professor Lecturer

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

May 2018

Table of Contents

[**Project Overview** 1](#_Toc513695568)

[**Need Analysis** 2](#_Toc513695569)

[**Objectives** 3](#_Toc513695570)

[**Literature Survey** 4](#_Toc513695571)

[**Methodology** 5](#_Toc513695572)

[**Requirement Analysis and Design Diagrams** 6](#_Toc513695573)

[**1.USE Case Diagram:** 6](#_Toc513695574)

[**2.Use Case Template:** 7](#_Toc513695575)

[**3.Activity Diagram:** 9](#_Toc513695576)

[**4.Sequence Diagram:** 10](#_Toc513695577)

[**5.Data Flow Diagram:** 10](#_Toc513695578)

[**6.Tasks and Subtasks:** 13](#_Toc513695579)

[**Assumptions and Constraints** 15](#_Toc513695580)

[**Analysis and Working Principles** 16](#_Toc513695581)

[**Design Model** 18](#_Toc513695582)

[**1. System Architecture:** 18](#_Toc513695583)

[**2. Context Diagram:** 18](#_Toc513695584)

[**3. MVC Architecture:** 19](#_Toc513695585)

[**4.Component Diagram:** 19](#_Toc513695586)

[**Work Plan** 21](#_Toc513695587)

[**Work Breakdown Structure** 22](#_Toc513695588)

[**Functional and Non-functional Requirements** 24](#_Toc513695589)

[**Project Outcomes** 27](#_Toc513695590)

[**Individual Roles** 27](#_Toc513695591)

[**COURSE SUBJECTS** 28](#_Toc513695592)

[**References** 29](#_Toc513695593)

# **Project Overview**

Engineering design problems are generally open-ended. They have no single correct answer, rather a range of possible solutions. The yearlong Capstone project keeps the desirability of open-ended design problems, the limitations of students’ knowledge and experience in check while helping us learn.

In the various laboratory classes in the college we have numerous viva(s) and quizzes. This task of conducting vivas and quizzes is quite cumbersome for the teachers as it takes a lot of preparation. Also, the checking process is quite redundant, time taking and tiring which leads to delayed results.

So, our team is trying to create a product “ViQu” which is a solution to the problem. A product based on Raspberry Pi 3 along with some sensors will be created, which will be able to conduct vivas, create quizzes and even check them hence, creating a digitized result sheet instantly.

Moreover, a database of MCQs and subjective viva-like questions will be created overtime by the teachers which will help in the future. This question bank will have ques of 3 difficulty levels.

Finally, the product ViQu will be able to conduct vivas, mark them, create quizzes, mark them and a fine database of questions will be created for the above.

# **Need Analysis**

* **Need to simplify the 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 are too tedious for the teachers. Hence, ViQu will take this responsibility off their shoulders.

* **Need for a question bank:**

Now a days the Quizzes need to be created by the teachers every time they are held. But with ViQu a large database of MCQs as well as subjective viva like questions will be created overtime which will help in the future.

* **Need for quicker and easier quiz checking process:**

As we see the process of checking quizzes is quite redundant, tedious and time taking hence, the results sometime get delayed. ViQu will simplify this process.

# **Objectives**

In order to fulfill our capstone project, we intend to achieve the following objectives:

* **To simplify the viva process:**

1. ViQu will ask the questions.
2. ViQu will create a transcript of the answer given by the student by using text summarization.
3. ViQu will do a probable marking and create a mark sheet.

* **To create a question bank:**

1. When the teachers will make the quiz the questions will be added to their question bank and by the passage of time it will improve.
2. ViQu will create quizzes and ask viva questions from these question banks.

* **To simplify the quizzing process:**

1. ViQu will generate the quiz based on the question bank.
2. ViQu will apply image processing to check the answered quizzes.
3. ViQu will evaluate these quizzes and will create a mark sheet.

* **To improve our skill set:**

1. We want to have a project experience in a team.
2. We would like to learn new technologies such as using Raspberry Pi, applying natural language processing.

# **Literature Survey**

Technology of automatic text summarization plays an important role in information retrieval and text classification, and may provide a solution to the information overload problem. Text summarization is a process of reducing the size of a text while preserving its information content. This paper proposes a sentences clustering based summarization approach. The proposed approach consists of three steps: first clusters the sentences based on the semantic distance among sentences in the document, and then on each cluster calculates the accumulative sentence similarity based on the multi-features combination method, at last chooses the topic sentences by some extraction rules.[1]

Text Summarization is the process of creating a condensed form of text document which maintains significant information and general meaning of source text. Automatic text summarization becomes an important way of finding relevant information precisely in large text in a short time with little efforts. Text summarization approaches are classified into two categories: extractive and abstractive. This paper presents the comprehensive survey of both the approaches in text summarization.[2]

Hiring processes, when conducted by humans at least, have always been problematic because bias is so often unconscious. [Personality](https://www.theguardian.com/science/2016/sep/01/how-algorithms-rule-our-working-lives) and [psychometric testing](https://www.theguardian.com/money/shortcuts/2014/jan/29/psychometric-tests-job-interviews-what-for), [blind auditions](https://www.theguardian.com/women-in-leadership/2013/oct/14/blind-auditions-orchestras-gender-bias), [webcam interviews](http://www.chicagotribune.com/business/ct-careers-webcam-job-interview-20161017-story.html) and [nameless CVs](http://uk.businessinsider.com/silicon-valley-bank-removing-names-cvs-resumes-prevent-unconcious-bias-2016-11?r=US&IR=T) 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](http://onlinelibrary.wiley.com/doi/10.1111/j.1468-0475.2011.00538.x/abstract?deniedAccessCustomisedMessage=&userIsAuthenticated=false). Humans are inconsistent where robots are incapable of being anything but consistent.[3]

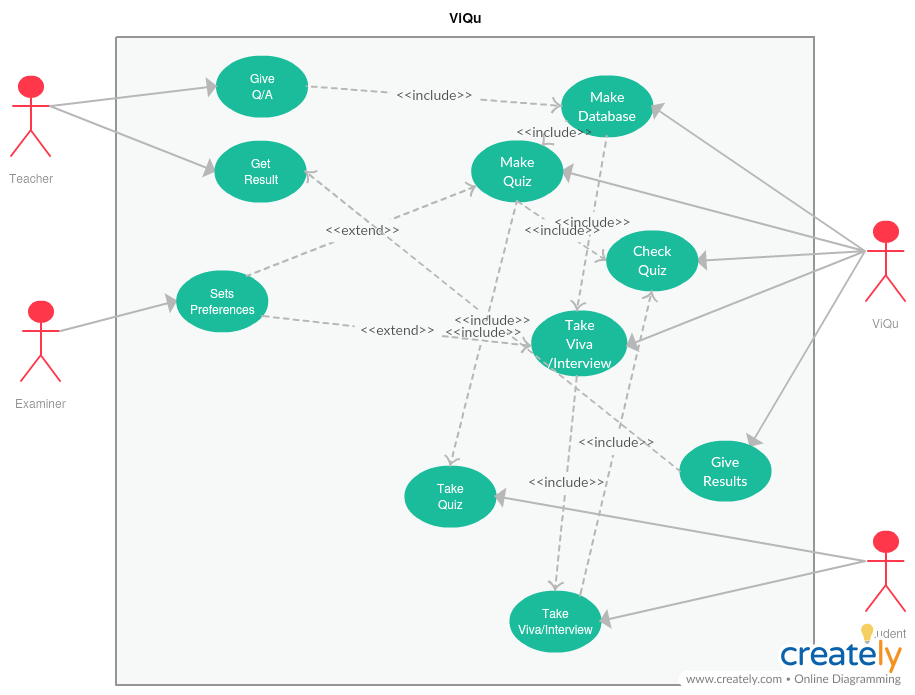
OCR reading technology is benefited by the evolution of high-powered desktop computing allowing for the development of more powerful recognition software that can read a variety of common printed fonts and handwritten texts. But still it remains a highly challenging task to implement an OCR that works under all possible conditions and gives highly accurate results.[4]

# **Methodology**

* We will be collecting the problems faced in the viva and quiz taking process and will also evaluate how the whole process is undertaken. Will properly analyze how the quizzes are set and how the marking schemes of viva and quizzes is set.
* Then we will be searching for the perfect solution through research about how these problems can be solved such as the use of Raspberry Pi, NLP, various APIs, etc. We will firstly try to test these individual modules on our own system before putting them to test on the actual product.
* Then we will go on with the proper hardware implementation and side by side create modules for various features.
* Now we will run certain tests and try to keep the scenario as real as possible. We will involve a few teachers in it who might help us in getting a small question database to be created.
* Finally, we will try to solve the errors which are seen in the previous stages and will further improvise/optimize the product as needed.
* Now we will put the product to real time testing and will also create a report on the same.

# **Requirement Analysis and Design Diagrams**

## **1.USE Case Diagram:**

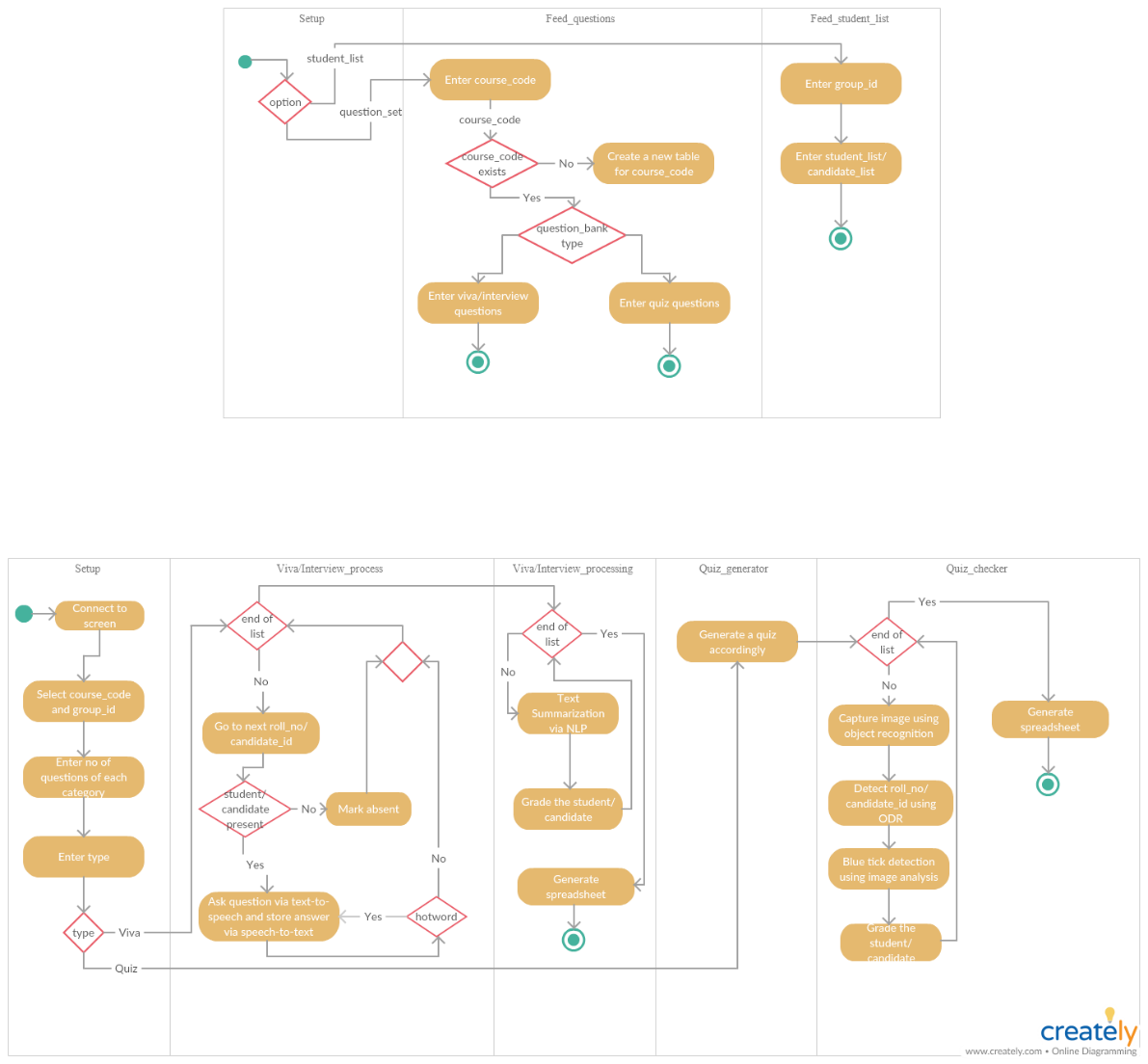


## **2.Use Case Template:**

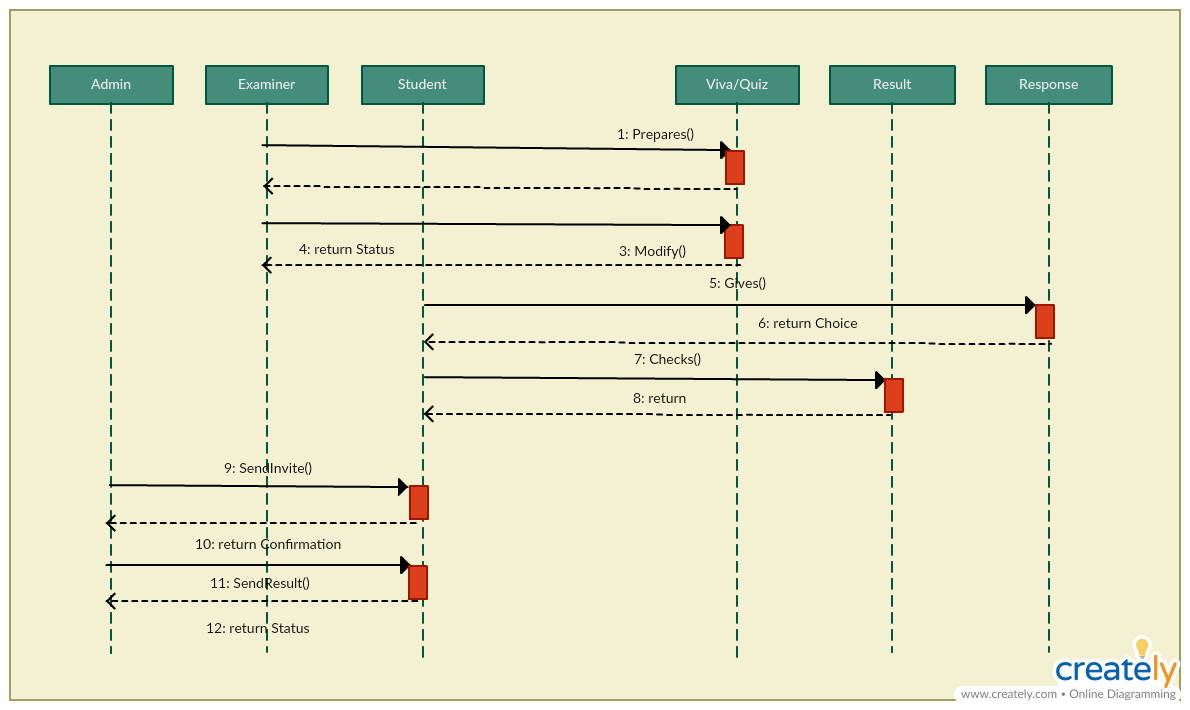
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UID-101 | | |
| Use Case Name: | Interviewing/Viva and Quiz system | | |
| Created By: | CPG28 | Last Updated By: | CPG28 |
| Date Created: | April 2nd, 2018 | Date Last Updated: | April 3rd, 2018 |

|  |  |
| --- | --- |
| 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. |

## **3.Activity Diagram:**

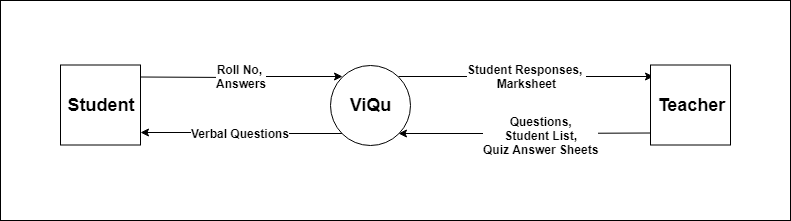


## **4.Sequence Diagram:**

****

## **5.Data Flow Diagram:**

**Level 0:**

****

**Level 1:**

**Level 2-1:**

****

**Level 2-2:**

****

## **6.Tasks and Subtasks:**

1. Identification of project
2. Planning of project and need analysis
   1. Planning
   2. Scope plan
3. Study
   1. NLP
   2. Image Analysis
   3. Raspberry Pi
4. Designing of various diagrams
   1. Use-case diagram
   2. Activity diagram
   3. Work Breakdown Structure
5. Hardware interfacing
   1. Setup of Raspberry Pi 3
      1. Setup sensors
         1. Camera
         2. Mic
      2. Setup Actuators
         1. Speaker
      3. Validating interfacing
6. Software interfacing
   1. Setup Module
      1. Create GUI (using Python – Tkinter library)
      2. Create Different modules for Viva/Interview and Quiz setup
   2. Viva/Interview Module
      1. Viva/Int erview Generator
         1. Question Bank Creation
         2. Question Selector based on difficulty
      2. Viva/Interviewing process module
         1. Text-to-speech module (for asking questions)
         2. Speech-to-text module (for registering response)
         3. Storage module
      3. Viva/Interview Processor
         1. Text-summarization module (using NLP)
         2. Keyword Extractor
         3. Grade Analyzer
         4. Spreadsheet Creator
   3. Quiz Module
      1. Quiz Generator
         1. Question Bank Creation
         2. Question Selector based on difficulty
      2. Quiz Analyzer
         1. Image Capturing module (using object recognition)
         2. Roll no./Candidate ID detection module (using ODR)
         3. Blue Tick Detection module (using Image analysis)
         4. Marking module
         5. Spreadsheet Creator

1. Integration
   1. Software and Hardware integration
   2. Module integration
2. Testing
   1. Unit testing
   2. Module testing
   3. System testing
   4. Acceptance testing
3. Deployment
   1. Beta testing
   2. Modifications
4. Evaluations

# **Assumptions and Constraints**

**Assumptions:**

1. There are sufficient number of questions of each level(easy/medium/hard) of both interview as well as quiz modules, along with answers.
2. An invigilator must always be present during the time of interview.
3. There is a specific place for the student's roll number on the quiz sheet.
4. The student checks the correct answer using Blue Ink pen on the quiz sheet.

**Constraints:**

1. At a given time only one teacher can work with the interview bot.
2. The interview bot should only be handed to a teacher, and not anyone else to ensure security.
3. The result sheet generated will not be saved by the interview bot once the bot is set up again.

# **Analysis and Working Principles**

**1. Deep Learning:**

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Deep learning models are loosely related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design, where they have produced results comparable to and in some cases superior to human experts.

We intend to use the same for text summarization process in the Interview/Viva process. Various open source libraries such as Tensor Flow, Microsoft Cognitive Toolkit, Theano, etc. work as an underlying principle to a Python library Keras, also open source, which we will be using to help the system in learning the text summarization patterns.

Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier.

Also, with further study we may use the same library for other computation heavy tasks such as ODR - Optical Digit Recognition and tick detection, both used in the quiz checking process.

**2. Raspberry Pi 3:**

To make the product a stand alone one we will be using the Raspberry Pi 3 single board computer which while being a simple and easy to use computer will also empower us to run the computationallyheavy 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.

**3. Speech-to-text and text-to-speech:**

These engines on the systems rely on the Google API which is available using the python library SpeechRecognition.

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.

**Design Model**

**1. System Architecture:**

Database generation

Send

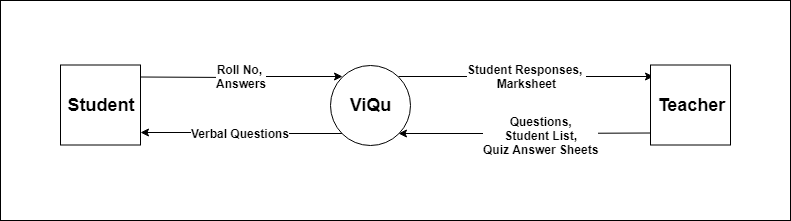
Results

Generate

Results

Interview/Quiz

**2. Context Diagram:**

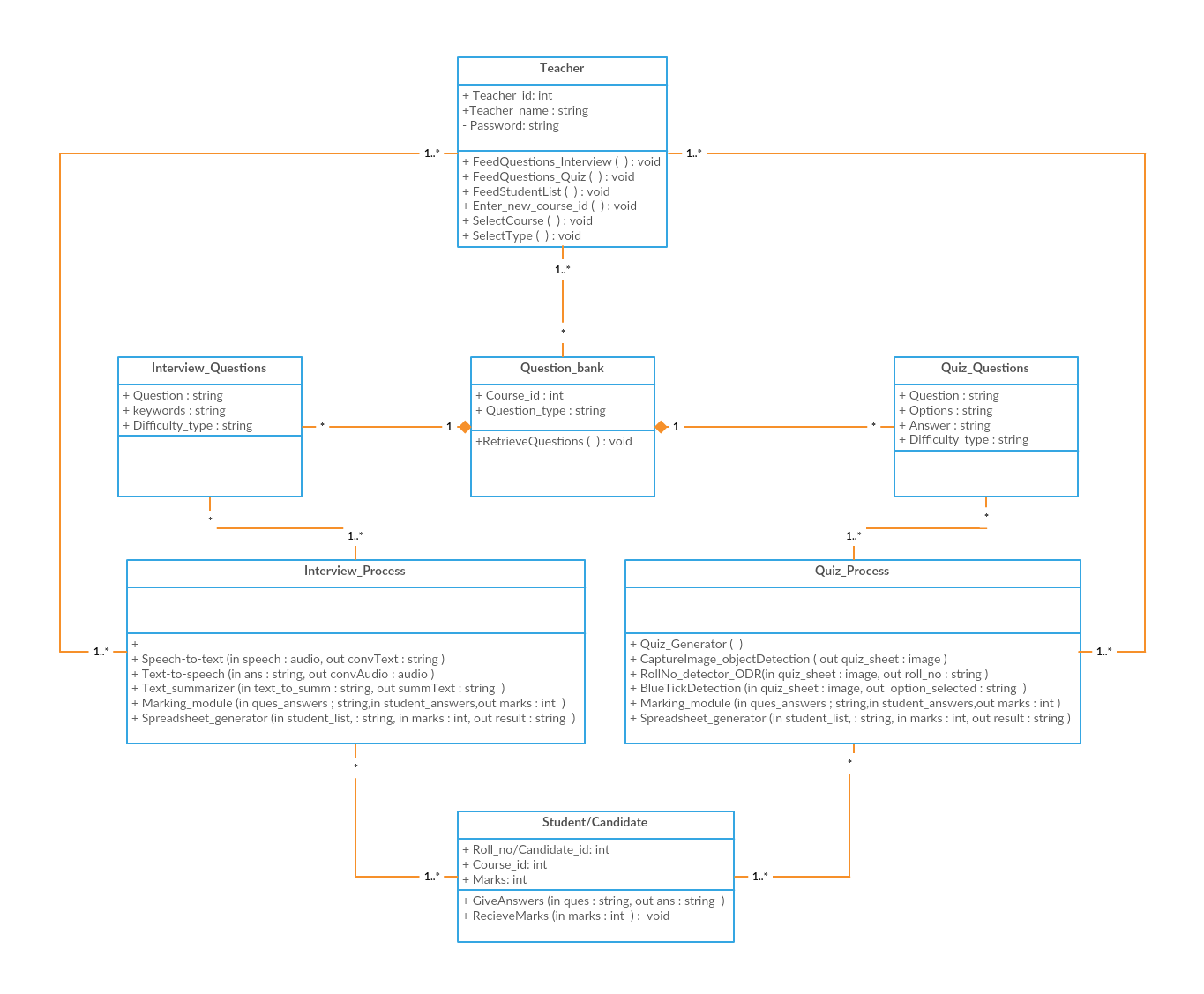
****

**3. MVC Architecture:**

Model-View-Controller Architecture

****

**4.Component Diagram:**

**5.Class Diagram:**

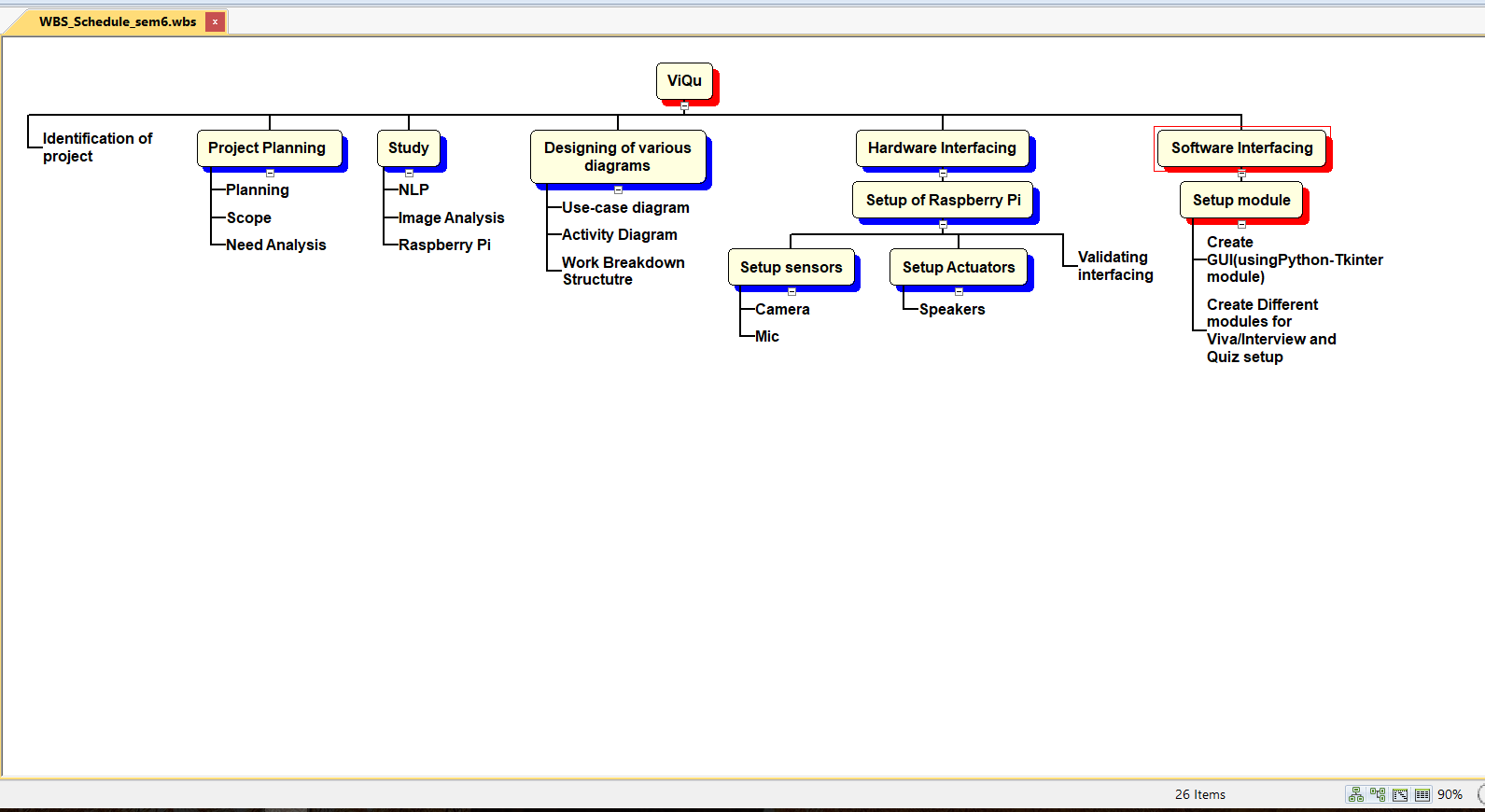
**Work Plan**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

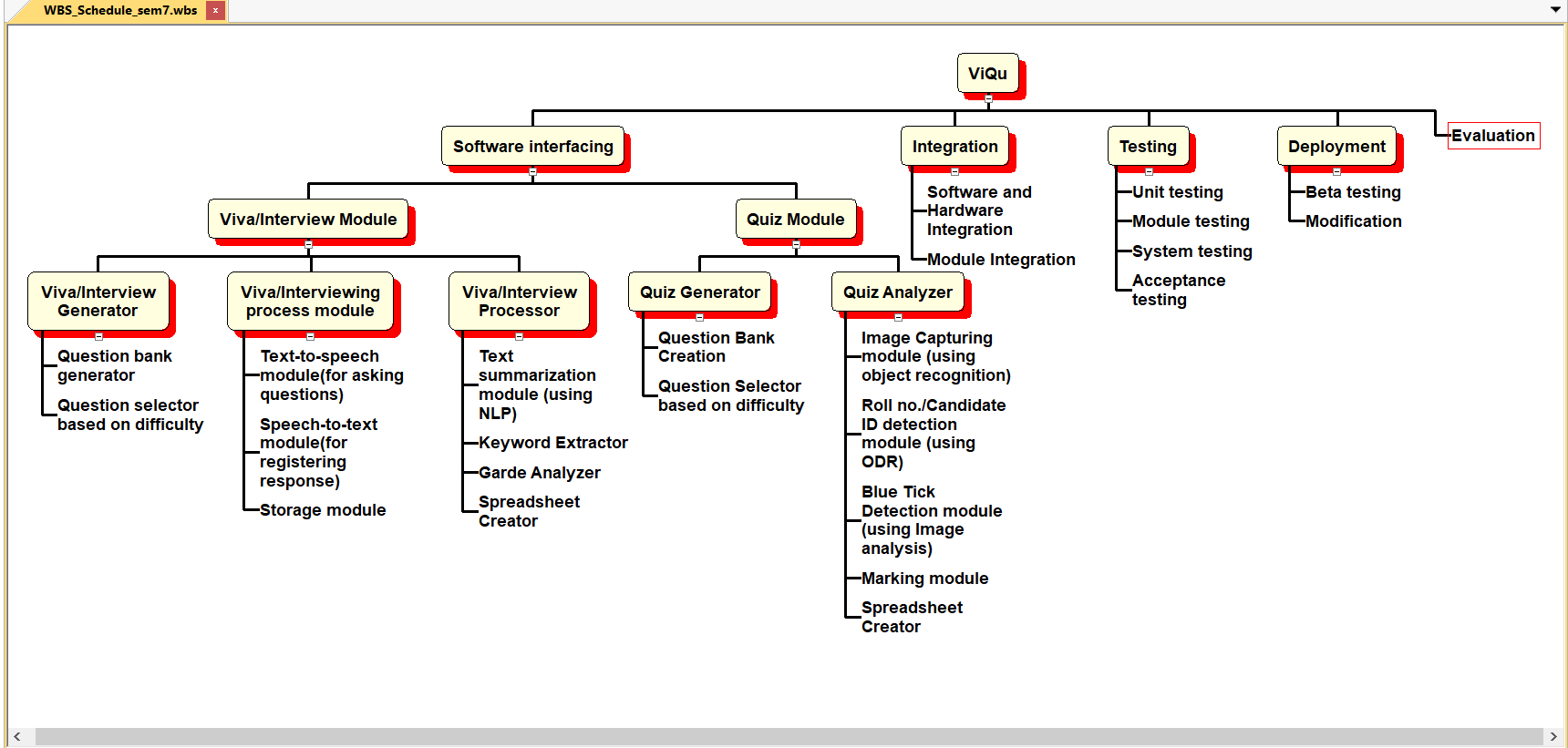
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Activity** | **Month** | **July** | **August** | | | | **September** | | | **October** | | | | **November** | | | | **December** |
|  |  | Week | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 5 | Software implementation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Design optimization |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Performing Modifications |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Results Evaluation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Final Report |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Work Breakdown Structure**

**Semester 6**

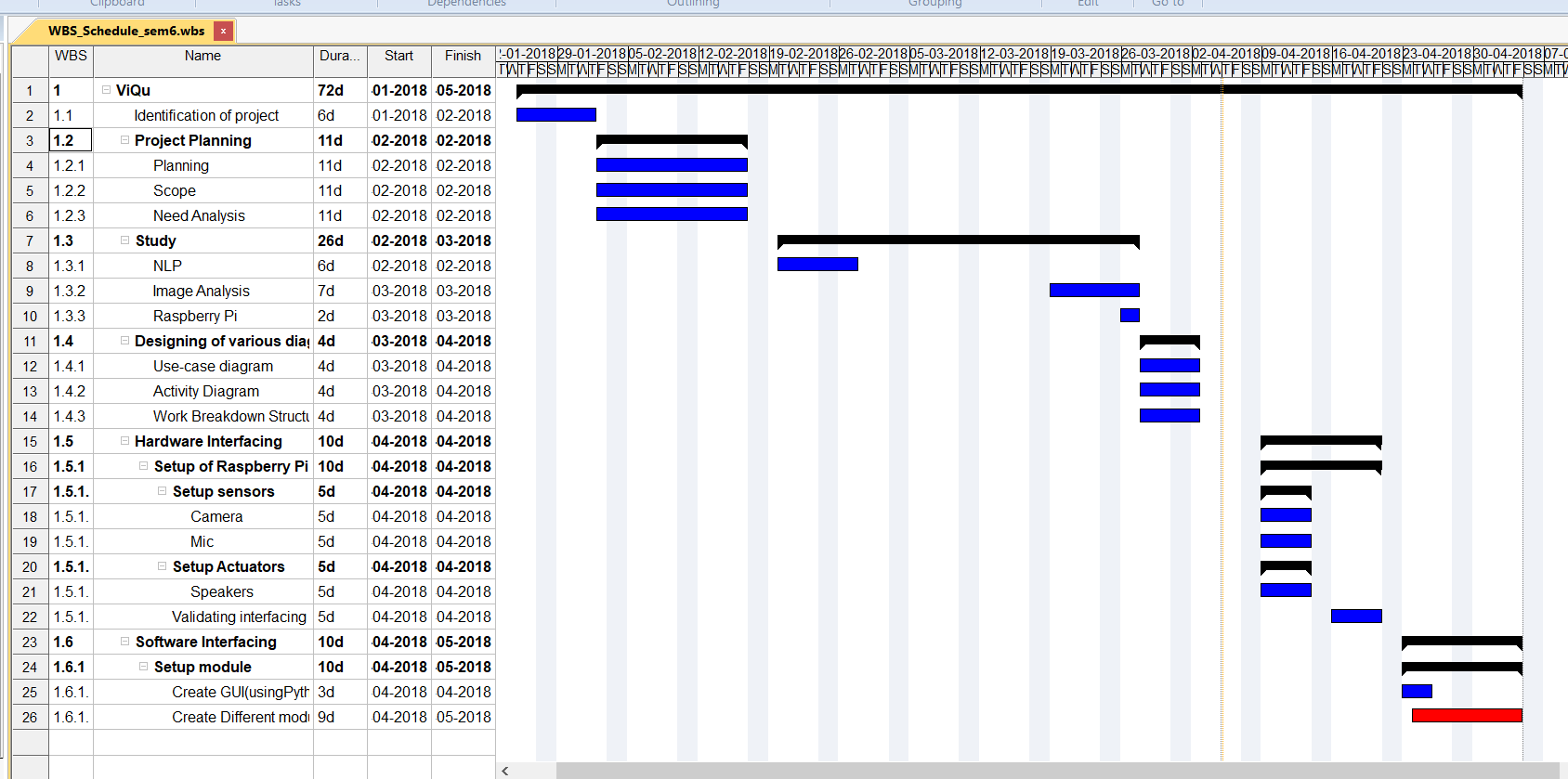


**Semester 7**

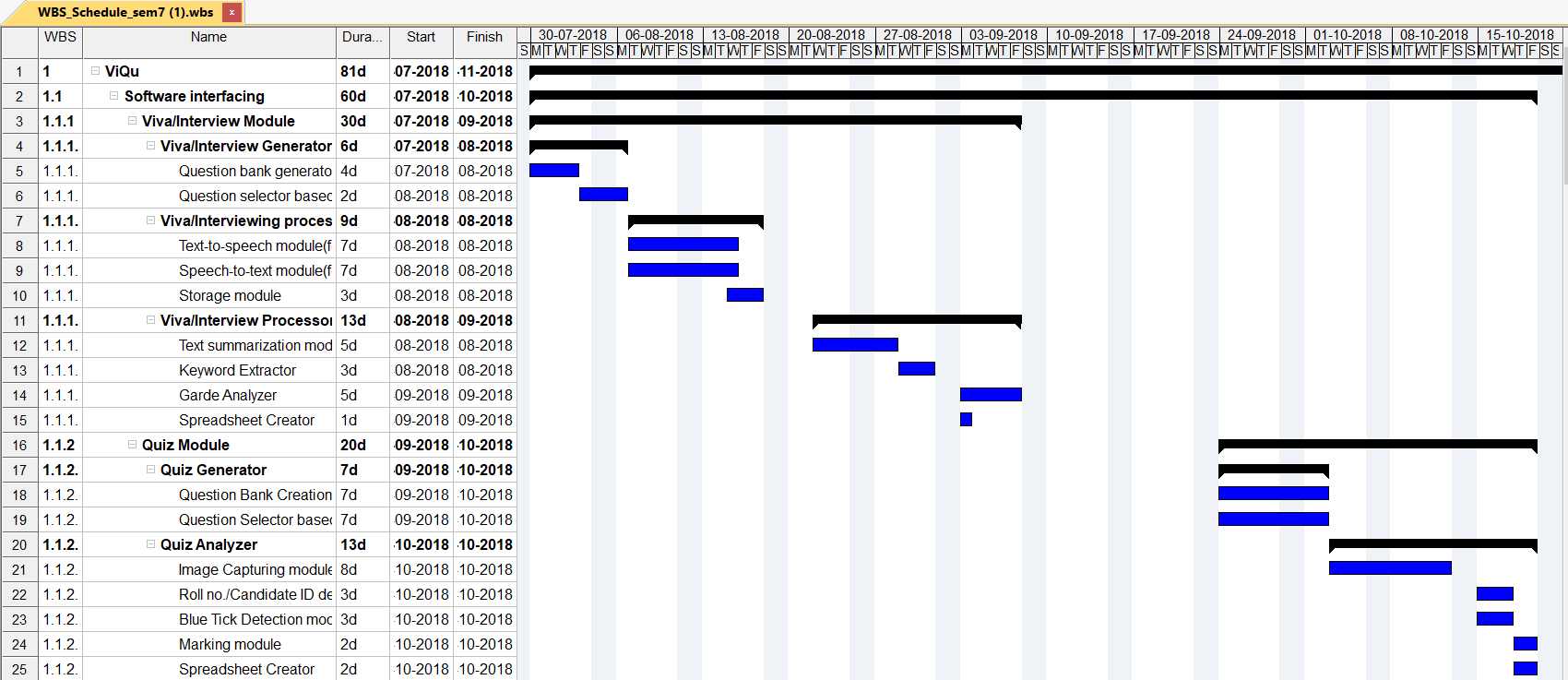


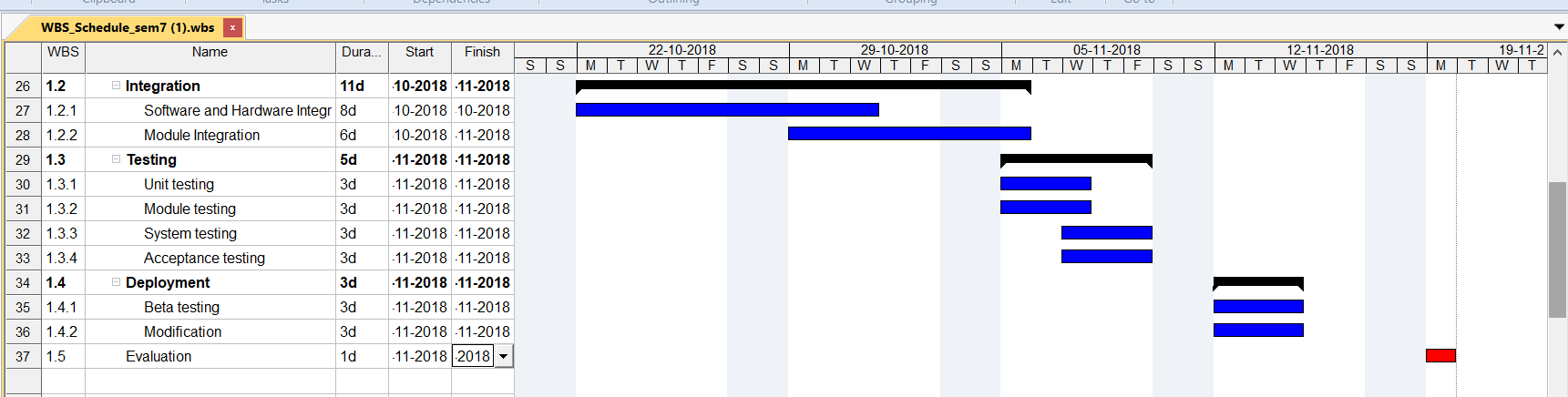
**Gantt Chart:**

**Semester 6**



**Semester 7**





# **Functional and Non-functional Requirements**

**Functional Analysis:**

Viva/Interview module:

* This module will fetch questions from the database to create a set of questions to be asked based on examiner preferences of difficulty level in the viva/interview.
* It will ask the candidate the set of questions created above using text-to-speech software module.
* It will record the response of the candidate to the asked question using speech-to-text software module.
* The recorded text above will then be summarized using the text-summarization software module.
* Also, the desired keywords will be searched in the candidate’s response.
* Based on certain parameters of marking such as number of keywords required and relevance to the actual answer, marks will be awarded on a scale of 5.
* Finally, when all the questions have been asked a summary of the viva/interview is created with the final score.
* Also, in case of multiple candidates a spreadsheet can also be created with the details of all the candidates.

Quiz module:

* This module will fetch questions from the database to create a set of questions to be asked based on examiner preferences of difficulty level in the viva/interview.
* A quiz will be generated according to the difficulty required.
* After the candidates have taken the quiz, the quiz sheet is scanned via the camera attached to the device.
* The module scans the roll number/candidate id from the sheet using ODR software module.
* Now it detects the blue colored ticks on the multiple-choice questions and evaluates it.
* Finally, it enters the score of the candidate in the spreadsheet.

Setup module:

* This module is used by the examiner to setup the device for various requirements of the viva/interview/quiz.
* The examiner can enter the number of questions to be asked.
* The examiner can enter the difficulty level of the viva/interview/quiz required.

**Non-Functional Analysis:**

**Performance requirement:**

* This is a hardware-based project, so the actual performance of the device is dependent on the hardware we choose to build it.
* Moreover, the project has various resource hungry software modules like text summarizer, ODR ,etc hence hardware is to be chosen careful.

Product Cost:

Since it is a hardware-based project various device must be bought to create it.

The estimated budget related to these devices are as follows:

1. Raspberry Pi 3 - ₹3150
2. Mic - ₹500
3. Camera - ₹1000
4. Speakers - ₹350

**Total** - **₹5000**

Hardware used:

1. Raspberry Pi
2. Mic
3. Camera
4. Speakers

Software used:

1. Python Language
2. Python libraries
   1. Tkinter
   2. Keras
   3. Pytessaract
   4. Text-to-speech library
   5. Speech-to-text library
3. OS for Raspberry Pi 3

Security:

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

Moreover, the device is not connected to the network, so it is not hackable.

# 

# **Project Outcomes**

Our project named “ViQu” is an assistant for taking viva, creating quizzes and analyzing them and evaluate accordingly and some the main objectives are as follows:

* ViQu will be a box case in which raspberry pi will be embedded with speakers, camera and mic.
* ViQu will be able to take the viva of a student and evaluate it, so as to make the process of viva faster and easier for both student and faculty.
* ViQu will also record these vivas for any future discrepancies if there are any.
* ViQu will also be able to create quizzes (MCQ type) and evaluate them by feeding the answered quiz picture to the ViQu and will finally give an excel file of the mark sheet.

**Individual Roles**

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

# 

# **COURSE SUBJECTS**

* Natural Language Processing – for summarizing text after speech recognition.
* Information Management System- for creating and maintaining our database and using SQL (structured query language)
* Image Processing- for recognizing MCQs based quiz sheets.
* Engineering Design- for implementation of Raspberry Pi (like Arduino)
* Python

**References**

[1] <http://ieeexplore.ieee.org/document/5234971/>

P. y. Zhang and C. h. Li, "Automatic text summarization based on sentences clustering and extraction," 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, 2009, pp. 167-170.  
doi: 10.1109/ICCSIT.2009.5234971

[2] <http://ieeexplore.ieee.org/document/7860024/>

N. Andhale and L. A. Bewoor, "An overview of Text Summarization techniques," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, 2016, pp. 1-7.doi: 10.1109/ICCUBEA.2016.7860024

[3]h<ttps://www.theguardian.com/careers/2016/dec/14/soon-robots-could-be-t>aking-your-job-interview

[4] <http://ieeexplore.ieee.org/document/5328096/>

M. A. Rahiman and M. S. Rajasree, "A Detailed Study and Analysis of OCR Research in South Indian Scripts," 2009 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, Kerala, 2009, pp. 31-38.  
doi: 10.1109/ARTCom.2009.45