Concept of Operations

Project Outcome:

This project will develop a 360-degree camera system for web conferencing, allowing clients to focus on the appropriate parties based on sound localisation. The task will be done by combining several physical devices and potentially existing open source libraries.

Project Scope:

In Scope	Out Of Scope
A 360 degrees web conferencing system Open source documentation to allow recreation of the product	Commercialisation of the product

Table 1. Project Scope

Stakeholders and Responsibilities:

Frederick Chew:

- Owner of the project
- Funding provider

360 Degrees Web Conferencing Team:

- Delivery of functional prototype
- Project documentation
- Handover document

Client's Requirements:

1. Powered by USB is preferable, if not then AC current or battery for portability.
2. The Raspberry Pi output the video in H264 video format. The video quality should be 1080p, if not 720p is acceptable.
3. Ability to reduce noise from external sources to avoid sound feedback.
4. The device does not cost more than $300 in total.
5. Plug and play with no extra software needed.
6. Combine multiple sources of videos and present multiple viewpoints at once.
7. The project should be open sourced.
8. It should be simple to build and operate. Proper documentation to ensure it is replicable
9. The device should be easy to upgrade in the future.

Requirements Analysis:

Requirement	Description	Ranking
R1	Portability	5
R2	Video Output	2
R3	Audio Output	5
R4	Cost	3
R5	Ease of Use	1
R6	Video Interface	2
R7	Simple Assembly	6
R8	Upgradability	4

Table 2. Requirements Ranking

Project Risks:

1. Failure of ReSpeaker 4-Mic Array
2. Delay in deliveries of components
3. DOA algorithm is not supported on the ReSpeaker 4-Mic Array
4. Video stream exceeds the Raspberry Pi USB bandwidth
5. 4 cameras set-up is not sufficient for 360o conferencing
6. Processing of videos cause input lag
7. Images and sounds are not synchronised.
8. Physical failures of components

Contingencies Plan:

Failure to Meet Deadlines	**Over Budget**	**Incompatible Integration**	**Damaged/Loss of Components**	**Delay Deliveries**
Inform other team members about problems in advance.	Notify the client the project is over budget	Discuss with members managing the different subsystems	Determine the fastest method to obtain replacement	Contact the seller for a delivery update
Discussion of a new deadline with team and client.	Discussion with other team members to see where cost can be reduced	Change the subsystems when necessary if cheaper to modify	Replace the component and notify the client and team of purchases	Purchase the component from a different seller
If unable to extend the deadline, the impact of failure must be considered.	Provide budget updates to client			Retain the other component as backup or resell to recover funds
Provide support to students to complete the task.				

Funding Expectations:

1. CECS Funding - $100
2. Client Funding - $200

Initial Project Timeline

Task	Description	Expected Completion
Initial components acquisition	Order the initial products such as raspberry-pi, microphone array and webcam	Week 3
Obtaining funding	Receiving microgrant from CECS and client	Week 4
Conceptual prototype	Develop a conceptual prototype with technical drawings for visualisation	Week 4
Methodology	Develop an initial methodology for tackling the problem	Week 4
Direction of arrival (DOA) algorithm	Developing and testing DOA algorithm for sound localisation	Week 4
Multiple camera setup	Setting up multiple video inputs on the Raspberry Pi	Week 5
Video and sound synchronisation	Ensure the sound and video inputs are synchronised with one another.	Week 6
Video switching	Switching camera input based on sound localisation	Week 6
Noise cancellation	Filtering out external noises and interferences from sound inputs	Week 7
Interface design	Design a unified interface for video output template from Raspberry Pi	Week 7
System optimisation	Optimise the data transmission between Raspberry Pi and computer to reduce overhead	Week 8
Prototype assembly	Assembling the final prototype and ensure the design is appropriate	Week 8
Prototype testing	Conduct testing of the 360o web conferencing device	Week 10
Handover document	Compilation of handover documents and ensure a guide is create for future assembly	Week 11

Table 3. Initial timeline

IP Consideration:

GNU GPL will be applied to the project to ensure it is open sourced

Sub-teams Responsibilities:

Sub-teams	Team Members
Sound (Direction of Arrival)	John, Jose
I/O (File Conversion)	Link, Jordan, Ben
Image (Processing)	Link, Jordan, Ben
Assembly (Modelling/Manufacturing)	Minh, Jireh
Documentation	Minh, Jireh

Table 4. Project sub-teams

Goals:

Baseline	Stretch Goals
• Camera switching based on sound input • Output of multiple videos as one source • Combining sounds and videos input into a unified output	• Real time, HD resolution 360 degrees web conferencing camera • Facial recognition • Image Stitching to form singular 360 degrees view

Table 5. Project goals

	Sound	Image	I/O	Assembly
Audit 1	Sound localisation algorithm	Multiple video inputs	Analysis of I/O standards	Acquisition of components
Audit 2	Input clarity Synchronisation with video input	Video switching based on sound inputs	Unified standards	Combining multiple cameras with sound system. Interfacing microphone array with raspberry pi.
Audit 3	Noise cancellation	Image stitching (Interface design)	Optimisation of data transmissions	Produce prototype housing

Table 6. Audit milestones

Concept Design:

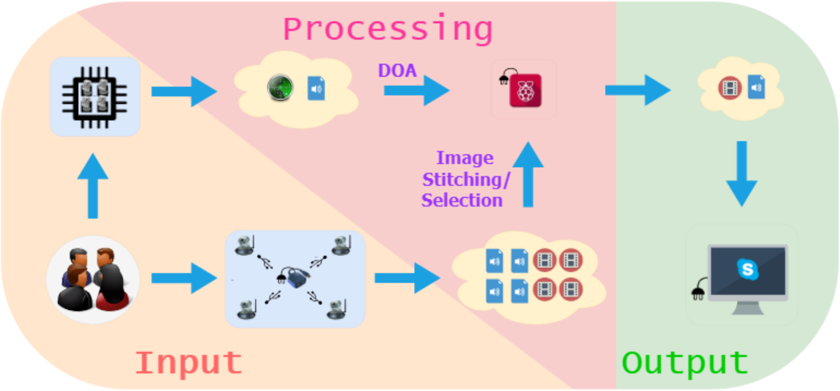


Figure 1. Concept design flow chart

Work Breakdown Structure:

Fully functioning 360o web conferencing camera system:

1. Buy cameras, Rasp Pi, and any hardware
2. Develop input subsystem
   2.1 Write a program to import images from multiple cameras
   2.2 Write a program to import sound from the microphones
3. Develop camera switching subsystem
   3.1 Write a program that is able to choose manually which camera for display

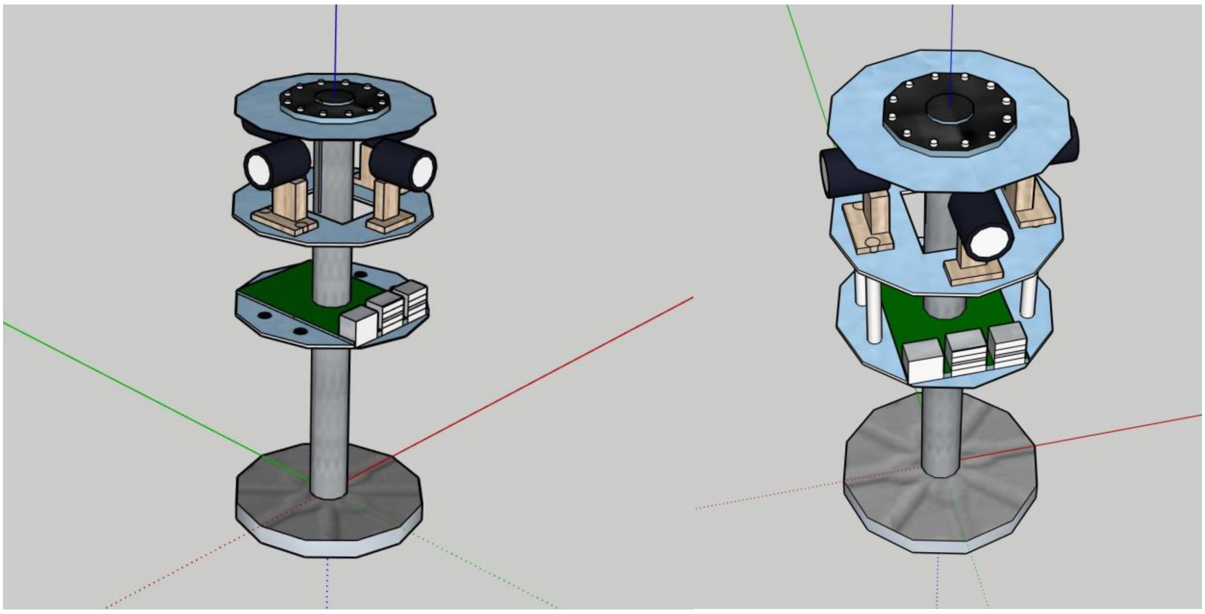
    3.2 Write a program that utilise the sound information to determine from which the sound is coming
    3.3 Write a program that associate the cameras and its corresponding direction

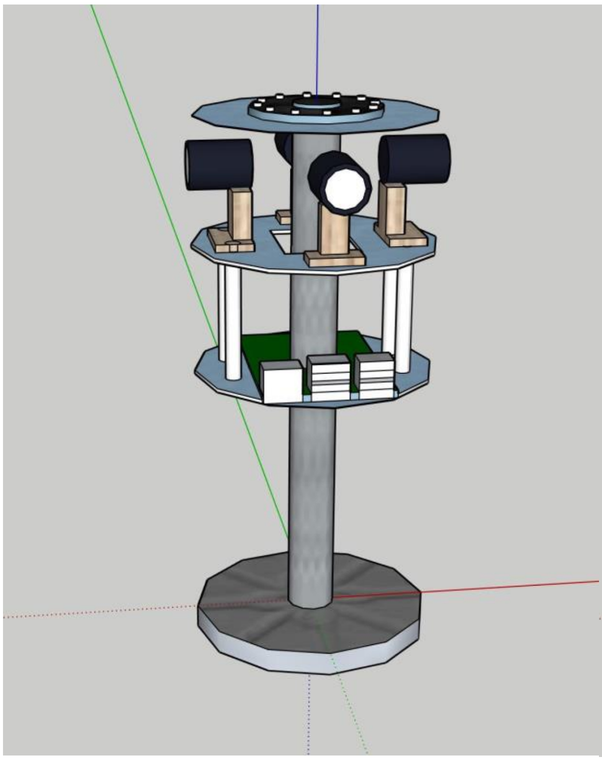
4. Develop the output subsystem
    4.1 Write a program that is able to generate the required video output given video in the program (so that the prototype works as a webcam for any computer software)
    4.2 Write a program that is able to combine the images and the audio information into video format
5. Assemble these subsystems into a prototype
    5.1 Design the prototype to be assembled given dimensions of all its components
    5.2 Design an efficient way of assembling all its components

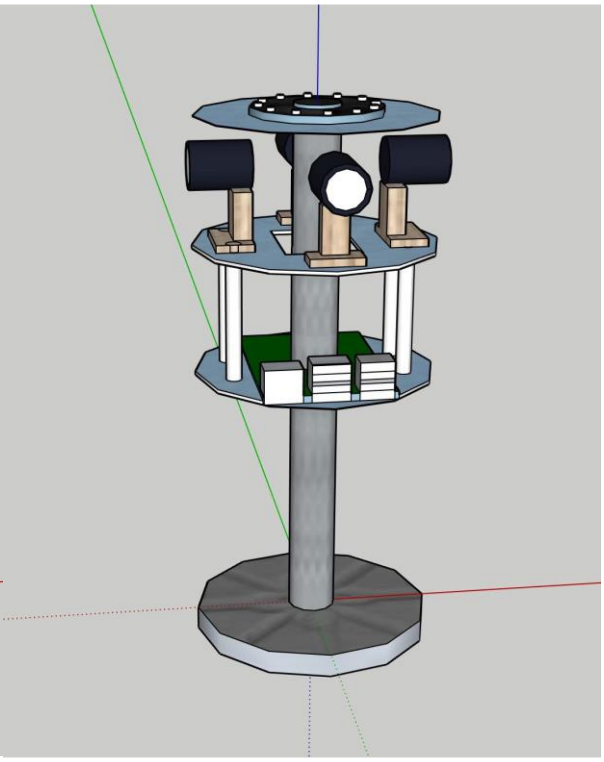
Work package index	1	2.1	2.2	3.1	3.2	3.3	4.1	4.2	5.1	5.2
Human resource (no. of people)	1	1 SE	1 SE	1 SE	1 SE	1 SE	1 SE	1 SE	1 SE	1 SE
Time resource (hours)	7.5	22.5	22.5	22.5	7.5	15	30	15	22.5	15
Object resource	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table 7. Work breakdown structure and resource allocation estimation

Initial Model Design:







Signatures:

Name: ________________________________ Signature: ________________________________

Name: ________________________________ Signature: ________________________________

Name: ________________________________ Signature: ________________________________

Name: ________________________________ Signature: ________________________________

Name: ________________________________ Signature: ________________________________

Name: ________________________________ Signature: ________________________________

Name: ________________________________ Signature: ________________________________

Name: ________________________________ Signature: ________________________________

Name: ________________________________ Signature: ________________________________