

ACKNOWLEDMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So, with gratitude, we acknowledge all those guidance and encouragement crowned our effort with success.

First and foremost, we would like to thank his Holiness Jagadguru Sri Shivarathri Deshikendra Maha Swamiji and **Dr. Bhimsen Soragaon**, Principal, JSSATE Bengaluru, for providing an opportunity to present this project as a part of my curriculum in the partial fulfilment of the degree course.

We express our sincere gratitude for **Dr. P. B. Mallikarjun**, Professor & Head, Department of Computer Science and Engineering, for his co-operation and encouragement at all moments of my approach.

We are sincerely grateful to our project guide **Ms. K V Shanthala**, Assistant Professor, Department of Computer Science and Engineering, for her spirited guideline and advice in carrying our project work and her valuable suggestions and her constant supervision has been very helpful.

We would like to thank the department for the constant encouragement, valuable help and assistance in every possible way. We would like to extend our sincere thanks to all the staff members for wholehearted support and co-operation.

Kiran A 1JS19CS076

Mudasir Ahamed 1JS19CS091

Nikhil Raju 1JS19CS102

Prithviraj Patil 1JS19CS125

ABSTRACT

Virtual reality, robotics, and Augmented reality can team up to develop innovative applications for various organizations. In this project a robot with a camera is placed in a remote location to capture the environment in visual form using Raspberry Pi (RPi). The captured visuals are displayed on the user's virtual reality (VR) headset. An added feature allows the camera to move in the direction of the user's head movements. This gives the user a real time experience as if he is present where the virtual tele-presence robot is located. The virtual telepresence robot can also be moved in any direction through an app installed in the user's smartphone.

TABLE OF CONTENT

Chapter No.	Chapter Name	Page No.
1	Introduction	1
	1.1 Virtual telepresence	1
	1.2 What is virtual telepresence robot?	1
	1.3 Existing system	1
	1.4 Proposed system	2
	1.5 Objectives	3
	1.6 Applications	3
2	Literature Survey	4
	2.1 Embedded System	7
	2.2 Applications of embedded system	9
3	Hardware and Software requirements	10
	3.1 Hardware requirements	10
	3.2 Software requirements	10
	3.3 Hardware description	11
	3.4 Software description	13
4	System Analysis	15
	4.1 System analysis	15
	4.2 Use case diagram	15
	4.3 Data flow diagram	16
	4.4 Functional requirement	17
	4.5 Feasibility study	17
5	System design	19
	5.1 Architecture diagram	19
	5.2 Sequence diagram	20
	5.3 Class diagram	21

6		Implementation	23
	6.1	Raspberry pi setup	23
	6.2	Bluetooth module setup	26
	6.3	Wi – fi module setup	26
	6.4	Power supply setup	26
	6.5	Code for robot movement	27
	6.6	Object detection code using flask	28
	6.7	Code for servo motors movement	29
7		System testing	30
	7.1	Testing for robot movement controls	30
	7.2	Testing of video streaming according to user's VR headset	30
	7.3	Testing of power supply	31
	7.4	Testing of voltage regulator	31
	7.5	Testing of raspberry pi	32
8		Results	33
9		Conclusion	39
	9.1	Conclusion	39
	9.2	Future enhancement	39
		References	40

LIST OF FIGURES

Figure No.	Figure Name	Page No.
2.1	Robot	9
3.1	Raspberry pi pin description	11
3.2	Raspberry pi camera	12
3.3	HC – 05 Bluetooth	12
4.1	Use case diagram	15
4.2	Initiation of process	16
4.3	Capturing live video	17
5.1	Architecture diagram	19
5.2	Sequence diagram	20
5.3	Class diagram	21
6.1	Installation of Raspberry pi os	25
6.2	Raspbian desktop	25
8.1	Front view of robot	33
8.2	Top view of robot	33
8.3	View of camera and raspberry pi modules	34
8.4	Picture captured during robot movement	34
8.5	VR headset	35
8.6	Bluetooth Terminal HC – 05 Application	35
8.7	Network analyzer applicaton	36
8.8	View of IP address in network analyzer	36
8.9	Wireless IMU application	37
8.10	Visuals of camera in dual browser application	37
8.11	Object detection view	38