

Robot-Assist

Big Data Analytics and Applications Project Proposal Document

Project Group #10

Sarika Bommavaram #10

Vaishnavi Aienampudi #01

Gouri Priya Vangavargu #45

Phanideep Salapaka #40

**Project Goal**

We can do many things with the help of robots. They can help people in many fields like doing some household work, helping in medical field, technological field etc.. The idea of this project comes from the question “Can a robot be a personal assistant to a disabled person?” We believe that humans can do this work but at some point of time or in some cases they become lazy or may not be able to do some work. So if we think of a robot, it can help humans to complete their work. This motivated us to make our robot as a personal assistant for a disabled person.

The goal of our project is to develop an application for a robot that helps disabled person in many ways. The feature of being interactive adds more importance to our application. Firstly, it helps the disabled person to know the weather of a particular city. Our application uses open weather API and text to speech API for this feature. It takes city name as input and gives weather details as output in speech form. Secondly, it helps the disabled person to make a call. It takes the command as call with some persons’ name and makes a call to that particular person. Thirdly, it also follows traffic directions.

**Increment #1**

For Increment #1, we have developed an application for Robo-Me that helps the disabled person in the following ways.

1) Helps the person to know weather of a particular city.

2) The Robot follows traffic signals.

3) The robot helps the disabled person to make a call to a particular person.

4) The robot can play music

5) The robot can determine the current location using Google Maps SDK.

6) Robot can click and save a photo to the library.

7) The robot can capture a person’s face and gives the confidence percentage whenever it detects his face.

**Application Specifications:**

|  |  |
| --- | --- |
|  | **Tools** |
| Platform | XCode |
| UML Diagrams | Microsoft Visio |
| Languages | Objective C |
| API’s Used | Weather API, Text-to-Speech API, Speech-to-text API, Google Maps SDK. |

**Existing Services:**

**1) Weather API:**

This API provides the service for weather. It takes input as name of the city and gives weather details like temperature, minimum temperature, maximum temperature etc…

<http://api.openweathermap.org/data/2.5/weather?q=Kansas>

**2) Text to Speech API:**

High-quality text-to-speech API. Adding parameter & return\_url=1 to the URL will give you the URL to generated MP3 instead.

<http://tts-api.com/tts.mp3?q=hello>

**3) Google Maps SDK:**

The use of google maps API is to know the current location. This is mainly used when user visits a new place, then user uses the API and determines his current location.

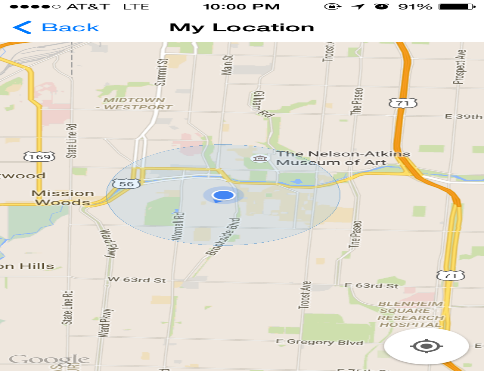
4) **Speech to Text**

We used Google’s speech-to-text API. It takes input as speech message and gives output as text message.

**Features for Increment #1**

For Increment #1, we have implemented the following features with RoboMe.

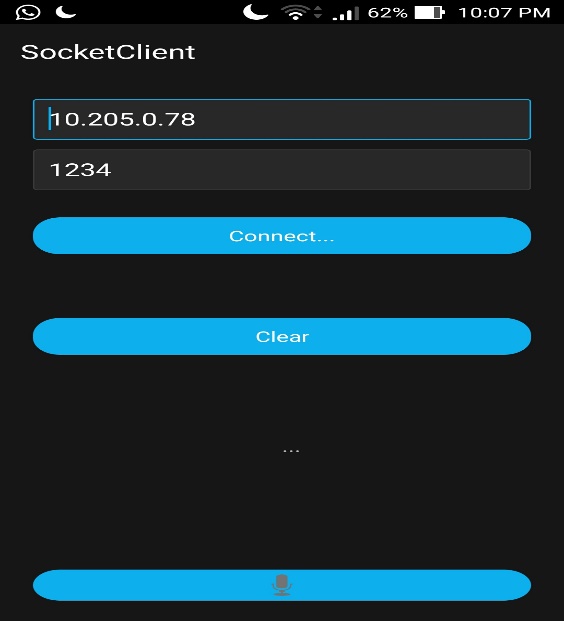
1. **Using Google Maps SDK**:

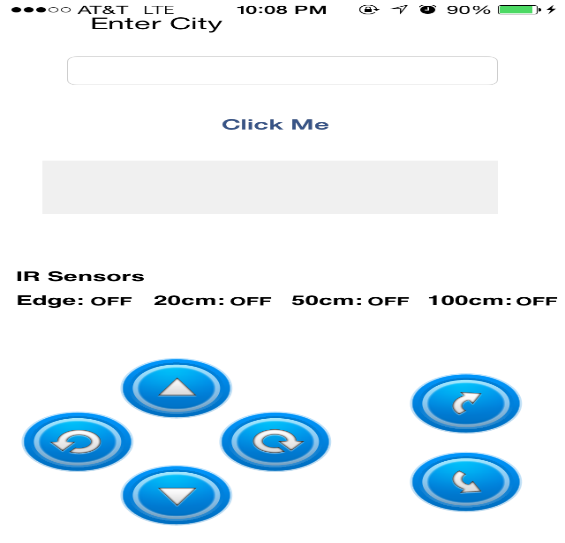


* With the help of Google maps SDK, we are determining the current location of the robot.

1. **Speak Weather:**

* This is the android controller where the user needs to enter the IP address of the robot.
* On click of connect it connects to the robot application.
* For this, we used open weather API, to get the weather conditions of a particular city. This feature will give result in a voice format. This is implemented by using text-to-speech API.
* The below screen shot shows the Robot interface on iPhone.

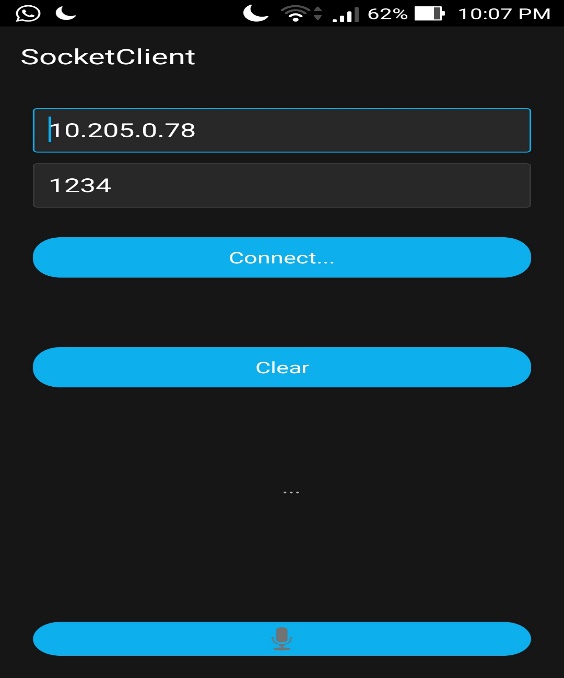
****

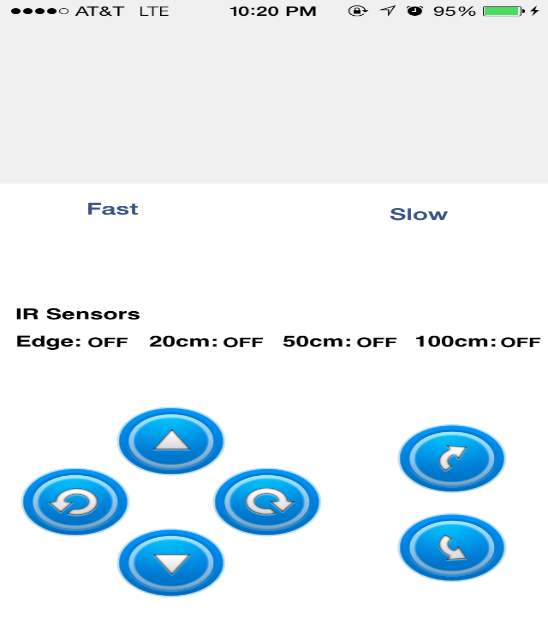


* The user enters the name of the city for which he wants to find the weather.
* On click of click me button, the user can listen the weather conditions.

1. **Robo - Me Commands:**

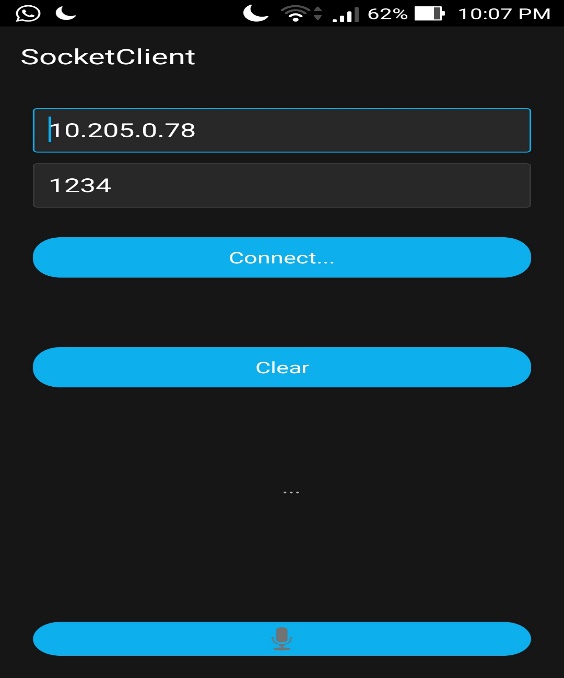
* This feature is implemented with the help of Robo me SDK. To control the movement of robot, we use our android device with Robo Me SDK as a controller. The controller takes input in the form of speech and the commands are passed to the robot.
* This screen shot shows the android controller where in user needs to enter the IP address and port number of the robo me application.
* The user can press the mic button and can give the commands to robot.
* The below screen shot shows the robo me application.
* If the user wants to move his robot at its maximum speed, he can click on fast button.
* If he wants the robot at its minimum speed, he can click on slow button.
* These are just optional features.

****

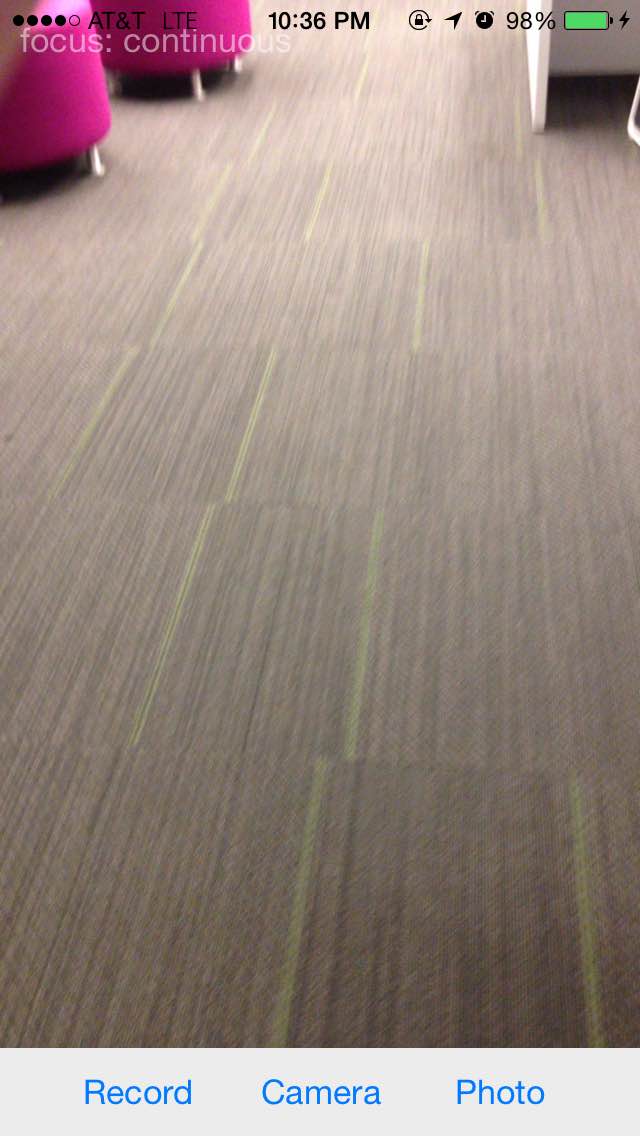


1. **Making a Call:**

* This is the android controller where the user can click on the mic button. If he gives command as call which is a voice text, the command will be received by the robo me application.
* On receiving the command “call”, a call will be automatically made to the emergency number.
* The user can also give command as “call name”.
* This feature is implemented with the help of Address book framework. It takes input as voice command from the android controller and makes a call to the desired person.

****

1. **Clicking a photo**

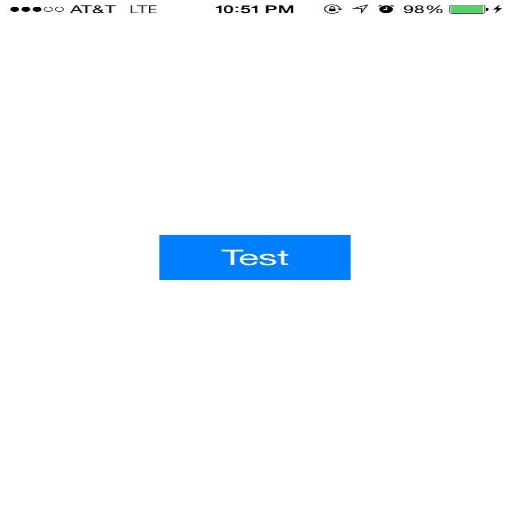


* This feature helps to click photos and save them in the iPhone photo library. This feature is implemented with the help AVCam framework.
* There are three buttons. If the user clicks on camera, he can switch between front and back cameras.
* Photo button is used to click photos and save them in our photo library in our device.
* Record button is used to record videos and save them to our device.

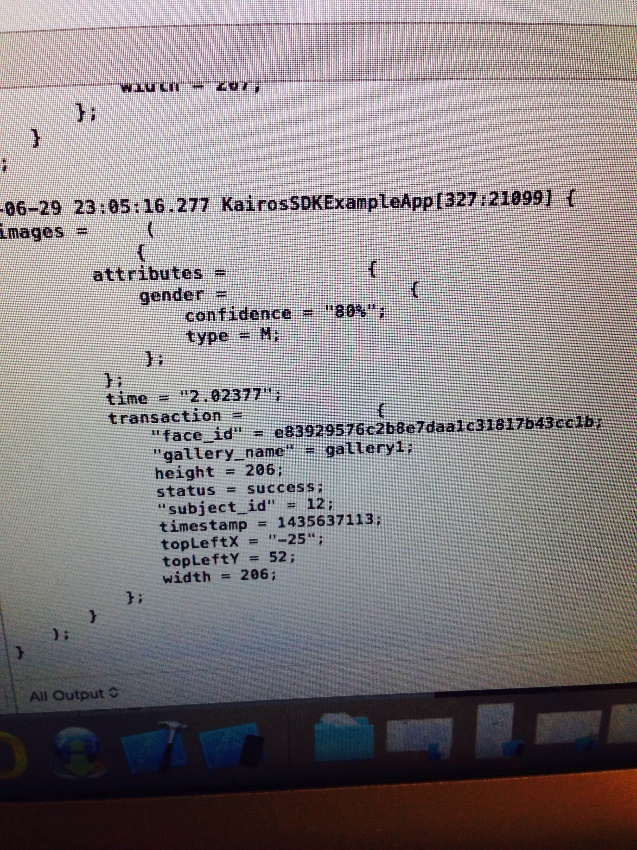
1. **Face Recognition**

* This feature will help the robot to capture the face of a particular person and whenever it detects that persons face, it gives the similarity percentage of the detected face. So, if the similarity score is greater than 60% it will greet that particular person. This feature is implemented with the help of image processing using Open CV.
* This screen shot shows the image capturing.
* Once, the image has been capured, the screen navigates to the next page, where there is a test button.
* On clicking the test button, it will match the captured image and the image it detected and return the confidence score.





****

****

* The figure shows the similarity score as 80% which shows the matching of images.

1. **Following traffic signals**

****

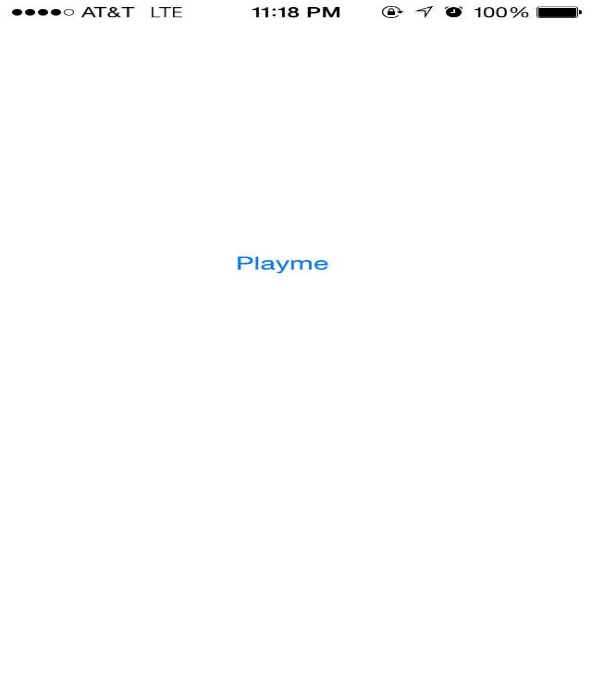
* From the figure, we can see the circle formed on detecting green color.
* We did this with the help of image processing using OpenCV.
* This concept is mainly used to follow traffic signals.
* We can see the below screen shot that forms circles on detecting red color.

****

* This feature helps the robot to follow the traffic signals. The robot will function in the following way:
  + If the robot detects a red color, it stops its movement.
  + If the robot detects a green color, it moves with its maximum speed.
  + If the robot detects an orange color, it moves with its minimum speed.

**8. Playing Music:**

* We implemented this feature of playing music on giving command “sing a song”.
* We used “AudioToolBox” framework for this feature.
* We can add music files and specify the path for the music file.



**References:**

* Tutorial for Objective C language - <http://www.tutorialspoint.com/objective_c/>
* Openweathermap API - <http://openweathermap.org/api>
* Text to speech API - <http://tts-api.com/>
* Googlemaps SDK - <https://developers.google.com/maps/documentation/ios/>
* OpenCV - <http://opencv.org/>
* RoboMe SDK - <https://github.com/WowWeeLabs/RoboMe-iOS-SDK>

**GitHub URL**

[https://github.com/vaishnavi5054/Big-data-Analytics-and-Apps/Increment #1](https://github.com/vaishnavi5054/Big-data-Analytics-and-Apps/Increment%20#1)