Living Requirements Document

Team: Voice++
Members: Petros Khachatrian (Lead), Chris Antanasian, Sheng Zheng, Julio Garcia, Belsin Barkhosir
Project name: InvocaBot
Revision History: Rev 1.0

Introduction:

InvocaBot is a hands-free agent for mobile devices and landlines that actively listens for a keyphrase made during a phone call. Upon a hearing a keyphrase, it executes the subsequent voice command. It receives the user’s voice commands through Sphinx-4 and our own software written in JRuby. InvocaBot will aid the user by implementing various services provided by the mobile device it is being used on and also other online services. A key feature of InvocaBot is that it is always listening; that is, there is no need to activate it through a keypress. This streamlines online and phone services while on a call.

Glossary of Terms:

FreeSwitch - a scalable open source cross-platform telephony platform designed to route and interconnect popular communication protocols using audio, video, text or any other form of media.

JRuby - JRuby is an implementation of the Ruby programming language atop the Java Virtual Machine, written largely in Java.

MGCP - Media Gateway Control Protocol: a standard protocol for handling the signaling and session management needed during a multimedia conference.

Phoneme - any of the perceptually distinct units of sound in a specified language that distinguish one word from another, for example p, b, d, and t in the English words pad, pat, bad, and bat.

Sphinx-4 - Adjustable and modifiable speech recognition system jointly designed by Carnegie Mellon University, Sun Microsystems laboratories, Mitsubishi Electric Research Labs, and Hewlett-Packard's Cambridge Research Lab.

Telecom Server - a server that allows us the enhanced capability to process incoming and outcoming audio through a phone call.
System Architecture Overview: (UML)

High-level Overview

Command Processing Workflow

"I have scheduled an appointment with Chandra at 12pm on Thursday, January 29th."
"Invocabot, Can you schedule an appointment with Chandra at 12pm this Thursday?"
"Invocabot, How far away is February 5th from today?"
"There are 7 days left until February 5th"
Requirements:

1. When the user begins a phone call, the system will connect to the user and begin listening:
   a. Give agent phoneme processing functionality.
   b. Configure agent to always be “listening” for correct pattern or phonemes to trigger it.
2. The user gives a command to the agent and awaits a response:
   a. Develop the agent to begin caching all phonemes until a decently sized pause in sound is detected.
   b. When a pause in sound is detected, make agent play confirmation blip.
3. The user hears the blip and awaits the agent to perform the given commands:
   a. Give agent ability to convert voice commands to text commands.
   b. Depending on command, perform functionality.
   c. Give agent the ability to respond with “Microsoft Sam - esque” voice.
   d. Agent should be able to interact with various applications if running on a mobile device, like Google Calendars, Notes, etc...
4. The user is finished with the agent and begins conversing with the other person on the line again:
   a. Make agent play different tone to signify functions are complete.
   b. Agent will return to “listening” mode after the “function-complete” tone plays.
5. When user is finished with a phone call:
   a. The agent should be closed when the call ends.
   b. The agent should output the summary of activities during the conversation when the call ends: which apps have been used and what has been edited.

   Example:
   - Call Duration: 23 min 23 sec
   - 12:20 PM Agent enabled.
   - 12:30 PM Notes: Party at 10 tonight.
   - 12:32 PM Googled “43rd street”.
   - 12:43 PM End call, agent disabled.

Prototyping Code + Test Cases:

Technologies Employed:
- Sphinx-4 Voice Recognition API
- JRuby 1.7.18
- Amazon Web Services (AWS)
- CloudOps private cloud