Software Requirements Specification

Speech Buddy

Speech Buddy Github

Version 1.0 approved

Prepared by William

The Arrested Developers

January 17, 2017

Table of Contents

Introduction

Proposal

An intelligent voice interface that is able to listen and interpret what the user has spoken. Input will be translated into text and stored in a Database.

The problem this project solves is that it helps users take simple notes, such as a grocery list or small reminders for when you don't have a pen and paper available. This project will help solved problems where people forget an important detail or appointment, by storing what the user says into a readable text format.

Similar to Apples Siri and Microsoft's Cortana.

Executive Summary

As students in the Computer Engineering Technology program, we will be integrating the knowledge and skills we have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a Database as well as to a mobile device application. The internet connected hardware will include a custom PCB with sensors and actuators for *speech recognition and voice recording*. The Database will store *what the user says in a readable format saved on a Database*. The mobile device functionality will include *storage for the recorded text and any reminders* and will be further detailed in the mobile application proposal. we will be collaborating with the following company/department, *we will not be collaborating with any companies at this moment*. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project.

Background

The problem solved by this project is that it helps users take simple notes, such as a grocery list or small reminders for when you don't have a pen and paper available. This project will help solved problems where people forget an important detail or appointment, by storing what the user says into a readable text format.

There are several applications and hardware out there that utilize speech as input, such as Apple's Siri, Microsoft's Cortana or Amazon's Alexa. Some other hardware and software similar to our projects are Digital pens. This piece of hardware records what users write down as input, and saves it in a text format on the computer. In this era there is not much mention of using a person's voice as input, and has not been a part a person's daily lives. You don't see people talking to their phone or microphone everywhere you look.

We have searched for prior art via Humber's IEEE subscription selecting "My Subscribed Content" and have found and read (Segura-Garcia, Felici-Castell, Perez-Solano, Cobos, & Navarro, 2015) which provides insight into similar efforts.

The first article contains information related to text-to-speech output in technology. (Karabetsos, Tsiakoulis, Chalamandaris, & Raptis, 2009)

The second article's information is about discriminating between vocal sounds and environment sounds. (Yuan-Yuan, Xue, & Bin, 2004)

The third article relates to the behaviour of speech with service robots. (Wang, Leung, Kurian, Kim, & Yoon, 2010)

In the Computer Engineering Technology program we have learned about the following topics from the respective relevant courses:

- Java Docs from CENG 212 Programming Techniques In Java,

- Construction of circuits from CENG 215 Digital And Interfacing Systems,
- Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
- Micro computing from CENG 252 Embedded Systems,
- SQL from CENG 254 Database With Java,
- Web access of Databases from CENG 256 Internet Scripting; and,
- Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:
Phase 1 Hardware build.
Phase 2 System integration.
Phase 3 Demonstration to future employers.

Phase 1 Hardware build

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

Phase 2 System integration

The system integration will be completed in the fall term.

Phase 3 Demonstration to future employers

This project will showcase the knowledge and skills that we have learned to potential employers.

The tables below provide rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

Labour Estimates	**Hrs**	**Notes**
Phase 1		
Writing proposal.	9	Tech identification quiz.
Creating project schedule. Initial project team meeting.	9	Proposal due.
Creating budget. Status Meeting.	9	Project Schedule due.
Acquiring components and writing progress report.	9	Budget due.
Mechanical assembly and writing progress report. Status Meeting.	9	Progress Report due (components acquired milestone).
PCB fabrication.	9	Progress Report due (Mechanical Assembly milestone).
Interface wiring, Placard design, Status Meeting.	9	PCB Due (power up milestone).
Preparing for demonstration.	9	Placard due.
Writing progress report and demonstrating project.	9	Progress Report due (Demonstrations at Open House Saturday, November 7, 2015 from 10 a.m. - 2 p.m.).
Editing build video.	9	Peer grading of demonstrations due.

Incorporation of feedback from demonstration and writing progress report. Status Meeting.	9	30 second build video due.
Practice presentations	9	Progress Report due.
1st round of Presentations, Collaborators present.	9	Presentation PowerPoint file due.
2nd round of Presentations	9	Build instructions up due.
Project videos, Status Meeting.	9	30 second script due.
Phase 1 Total	**135**	
Phase 2		
Meet with collaborators	9	Status Meeting
Initial integration.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Meet with collaborators	9	Status Meeting
Incorporation of feedback.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Prepare for demonstration.	9	Progress Report
Complete presentation.	9	Demonstration at Open House Saturday, April 9, 2016 10 a.m. to 2 p.m.
Complete final report. 1st round of Presentations.	9	Presentation PowerPoint file due.
Write video script. 2nd round of Presentations, delivery of project.	9	Final written report including final budget and record of expenditures, covering both this semester and the previous semester.
Project videos.	9	Video script due
Phase 2 Total	**135**	
Phase 3		
Interviews	TBD	
Phase 3 Total	**TBD**	
Material Estimates	**Cost**	**Notes**
Phase 1		
A microcomputer composed of a quad-core Windows 10 IoT core compatible Broadcom BCM2836 SoC with a 900MHz Application ARM Cortex-A7 32 bit RISC v7-A processor core stacked under 1GB of 450MHz SDRAM, 10/100 Mbit/s Ethernet, GPIO, UART, I²C bus, SPI bus, 8 GB of Secure Digital storage, a power supply, and a USB Wi-Fi adaptor.	\$120.00	Amazon
Microphone	\$20.00	
Speaker	\$28.99	
CENG Parts Kit	\$110.00	
J206 Parts	\$40.00	
Phase 1 Total	**>\$318.99**	
Phase 2		
Materials to improve functionality, fit, and finish of project.	N/A	
Phase 2 Total	**TBD**	
Phase 3		
Off campus colocation	<\$100.00	An example: [4].

Shipping	*TBD*
Tax	*TBD*
Duty	*TBD*
Phase 3 Total	

Concluding Remarks

This proposal presents a plan for providing an IoT solution for better planning and helps people set reminders of important details. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects. I request approval of this project.

Database Specifications

Database Type

Firebase is the Database type that is used for Speech Buddy. Firebase is a JSON style Database that is organized via encrypted keys. It is incorporated directly in android studio for ease of use with mobile applications wanting to store data online.

Database Tables

There are three tables that are used in the Firebase.

1. Users
2. ListNames
3. ItemNames

Users is the table in which all users who sign up for our Speech Buddy Application are stored. They are given a key as a unique identifier.

ListNames is the table in which the names of list the user created are stored as strings. Each Users own Lists that they have created will be unique to them via an identification key within the table.

ItemNames is the table in which the items under each list are stored. Each Item is unique to the list name it was created under and the user that created it via an identification key within the table.

Database Alteration via User Case Examples

User case: The User Adds a list named “groceries” to their speech buddy.

Consequences: A string element “groceries” and key unique to the user are generated in a new row under the ListNames table.

User case: The User Creates an Account with username steve@abc.com and password “123”

Consequences: A string element (in mandatory email format) “steve@abc.com”, a string password “123” and key unique to the newly made user are generated in a new row under the Users Table. All data manipulation under their account will use this identifier, and they must use these credentials to access the application.

User case: The User Adds an Item named “rice” to their “groceries” list.

Consequences: a string element “rice” and key unique to the user are generated in a new row under the ListNames table.

Mobile Application Specifications

Graphical User Interface Specifications

Login Activity: The login screen of the application is the first presented upon launching the application. The user may enter already existing credentials into EditText fields and login, or may choose to go to the sign up activity, both navigations via Buttons.

Sign Up Activity: The sign up screen of the application can be accessed from the login menu. It has two EditText fields for new credentials to be added, which upon submitting via Button will be verified by the username and password guidelines and added to the user database. Incorrect EditText entries will be communicated via Toast message asking for re-entry.

List Activity: The list activity of the application is comprised of a single list, formatted appropriately, displayed across the width of the screen in both portrait and landscape orientation. It loads it's elements from the database dynamically. If the list length exceeds the screen size, the list becomes scrollable. The top right has the dropdown navigation menu, and add list/delete list buttons.

Items Activity: The item activity of the application is comprised of a single list, formatted appropriately, displayed across the width of the screen in both portrait and landscape orientation. It loads it's elements from the database dynamically. The elements it loads are specific to the parent list and user selected. If the list length exceeds the screen size, the list becomes scrollable. The top right has the dropdown navigation menu, and add item/item list buttons.

About Application Activity: A simple activity describing the purpose and contributors of the Speech Buddy Application via formatted TextViews.

Dropdown Navigation Menu: In the top right corner of post-login activities. Allows the user to Navigate to list pages, the about page, and logout. Also has an option to clear all data from the current users records to start anew.

Additional Note: For ease of use, users may also press and hold list/item elements to delete them. Any deleted list will also delete it's child items.

Database Integration

The Speech Buddy Application uses the integrated Firebase libraries in Android Studio to connect to the database. Once connected online, it actively retrieves data and adds dynamically updates the lists displayed within the application. All new elements added to the users lists and items are independently added to the database as soon as they are entered for optimal up to date accuracy within the database.

User Cases

Test case: 1
Creating tables
Purpose: Purpose is to observe if our application has created a table that contains the columns. Columns we need are the ID and the name of all list the user created.
Precondition:
Steps:
Expected Result: Table should be created with an ID that auto increments, and content the names for each column

---

- Have created a database
- Make the columns need for table (Strings)
- CREATE table statement

1. Create a database using SQLite
2. Make column names
3. Create a String for the CREATE table
4. Run the statement using database.execSQL()
5. Go to where database saved and check if it is there
6. Open database see if table is there

Test case: 2
Insert data into Table
Purpose: Purpose is to observe if our application can allow users to enter data that will be stored in a database
Precondition:
Steps:
Expected Result: Table should be created with an ID that auto increments and the String the user entered in the EditTextView for the correct column

---

- Have created a database
- Have the data needed to insert into table
- ID (auto incremented)
- List name (user enters this)

1. Get data to enter (list name) from EditTextView
2. Save data in table made variables
3. Insert the data into the table one row at a time
4. Go to where database saved and check if it is there
5. Open database see if table is there

Test case: 3
Delete data into Table
Purpose: Purpose is to let users remove any old list they have or mistakes they have made
Precondition:
Steps:
Expected Result: The data enter by the user should be deleted from the table

---

- Have created a database
- Have the data you want to delete needed in the table

1. Get the name of the list/data user wants to delete from table
2. Create a DELETE statement using that name and the table name
3. Run the statement using database.delete()
4. Open database and table and see result

Test case: 4

Delete all data on database
Purpose: Remove all the data in the database if user wants a fresh restart with no data
Precondition:
Steps:
Expected Result: Database will contain tables with no data

- Have created a database
- Have tables with data in them

1. Go into setting activity and click button
2. Confirm yes or no
3. Run delete Statement for all data in columns
4. Tables will still exist

Test case: 5
Blank inputs
Purpose: See what will be stored in the database if user enters blank items
Precondition:
Steps:
Expected Result: A blank entry will be created in the database

---

- Have created a database
- Text box for user input

1. Go into mylists
2. Add a list will blank information
3. Open database to see result

Test case: 6
Delete using a blank input
Purpose: Observe what when user enters Blank, see how it affects the database
Precondition:
Steps:
Expected Result: Will look for a blank data in table and remove it, if it cannot find data do nothing

---

- Have created a database
- Text box for user input

1. Go into mylists
2. Delete a list will blank information
3. Open database to see result

Test case: 7
User hits the back key on application
Purpose: To test if the application will close if user clicks the back key
Precondition:

Steps:
Expected Result: Application should prompt user if they want to exit the application

---

- Run the application
- VM of android phone

1. Run the application
2. clicks the back key

Test case: 8
Delete using a blank input
Purpose: Observe what when user enters Blank, see how it affects the database
Precondition:
Steps:
Expected Result: Will look for a blank data in table and remove it, if it cannot find data do nothing

---

- Have created a database
- Text box for user input

1. Go into mylists
2. Delete a list will blank information
3. Open database to see result

Test case: 9
Enter a list with the same name as another
Purpose: Test to see if database will create two data entries that are the same
Precondition:
Steps:
Expected Result: If you click any of the listnames with the same name, the item data is the same for all of them

---

- Have created a database
- Enter data for both list and item tables

1. Enter 2 entries with the same listname
2. Click one of them and enter an item

Application Work Contributions

William: The acting scrum master of the project. William is in charge of back-end development including Java and using libraries and API's. He controls the overall architecture and functionality of the project.

Sanjay: In charge of the Graphical User Interface and User experience. Designs layouts and recommends functionality to William to be implemented.

Kevin: In charge of data storage, manipulation and maintenance. Created the Firebase database, as well as its internal structure and breakdown to be connected to via Firebase Java libraries.

Additional Web Specifications

Amazon Voice Services

The Speech Buddy Hardware utilizes Amazon Voice Libraries for voice input to be added to the database. In addition to the JavaScript classes built for Speech Buddy, Amazon Voice Services offers basic user searches via an online search engine and can answer simple questions. These library abilities are not implemented in Speech Buddy's Functionality.

Firebase Database

The Speech Buddy Hardware and application will be utilizing Firebase databases for data storage. The service offers free (to a certain amount of traffic) data hosting and an appropriate size and speed for Speech Buddy's Requirements.

References

Karabetsos, S., Tsiakoulis, P., Chalamandaris, A., & Raptis, S. (2009). Embedded unit selection text-to-speech synthesis for mobile devices. *IEEE Transactions on Consumer Electronics*, *55*(2), 613–621. https://doi.org/10.1109/TCE.2009.5174430

Segura-Garcia, J., Felici-Castell, S., Perez-Solano, J. J., Cobos, M., & Navarro, J. M. (2015). Low-cost alternatives for urban noise nuisance monitoring using wireless sensor networks. *IEEE Sensors Journal*, *15*(2), 836–844. https://doi.org/10.1109/JSEN.2014.2356342

Wang, D., Leung, H., Kurian, A. P., Kim, H. J., & Yoon, H. (2010). A deconvolutive neural network for speech classification with applications to home service robot. *IEEE Transactions on Instrumentation and Measurement*, *59*(12), 3237–3243. https://doi.org/10.1109/TIM.2010.2047551

Yuan-Yuan, S., Xue, W., & Bin, S. (2004). Several features for discrimination between vocal sounds and other environmental sounds. In *2004 12th european signal processing conference* (pp. 2099–2102).