EchoBudget

B0: Lynn Sun, Yuxuan Xiao, Yixin Yang
18-500 Capstone Design, Spring 2024
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

EchoBudget is a web-based application that provides money-tracking functionalities with audio input. Traditional money-tracking apps are not accessible for visually impaired people and the elderly due to a lack of visual aids and complex user interfaces. To meet their needs for money tracking, we propose to implement an application that records, shows, and analyzes spending solely with audio input and output.

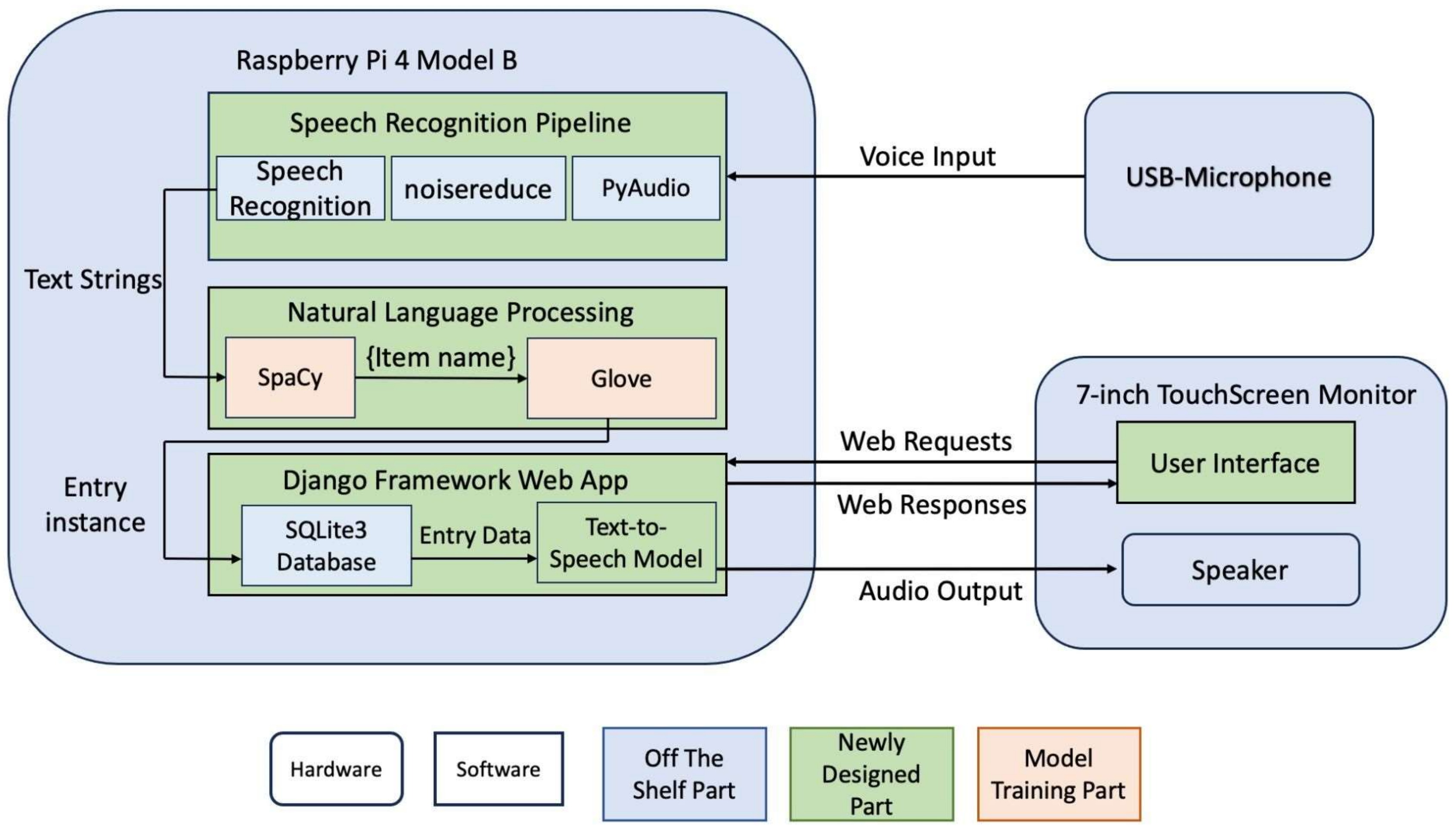
We successfully implemented full functionality for money tracker apps, allowing customers to use their voice to enter new entries and edit or remove existing entries. Our application could also display existing entries and generate reports of expenses for a given range depending on users' audio requests. Our whole system weighs 500 grams, making it easy for customers to carry around. The overall accuracy of the system is about 95 percent and it would take less than 10 seconds for any command to be processed, giving user a smooth interaction with our system.

System Architecture

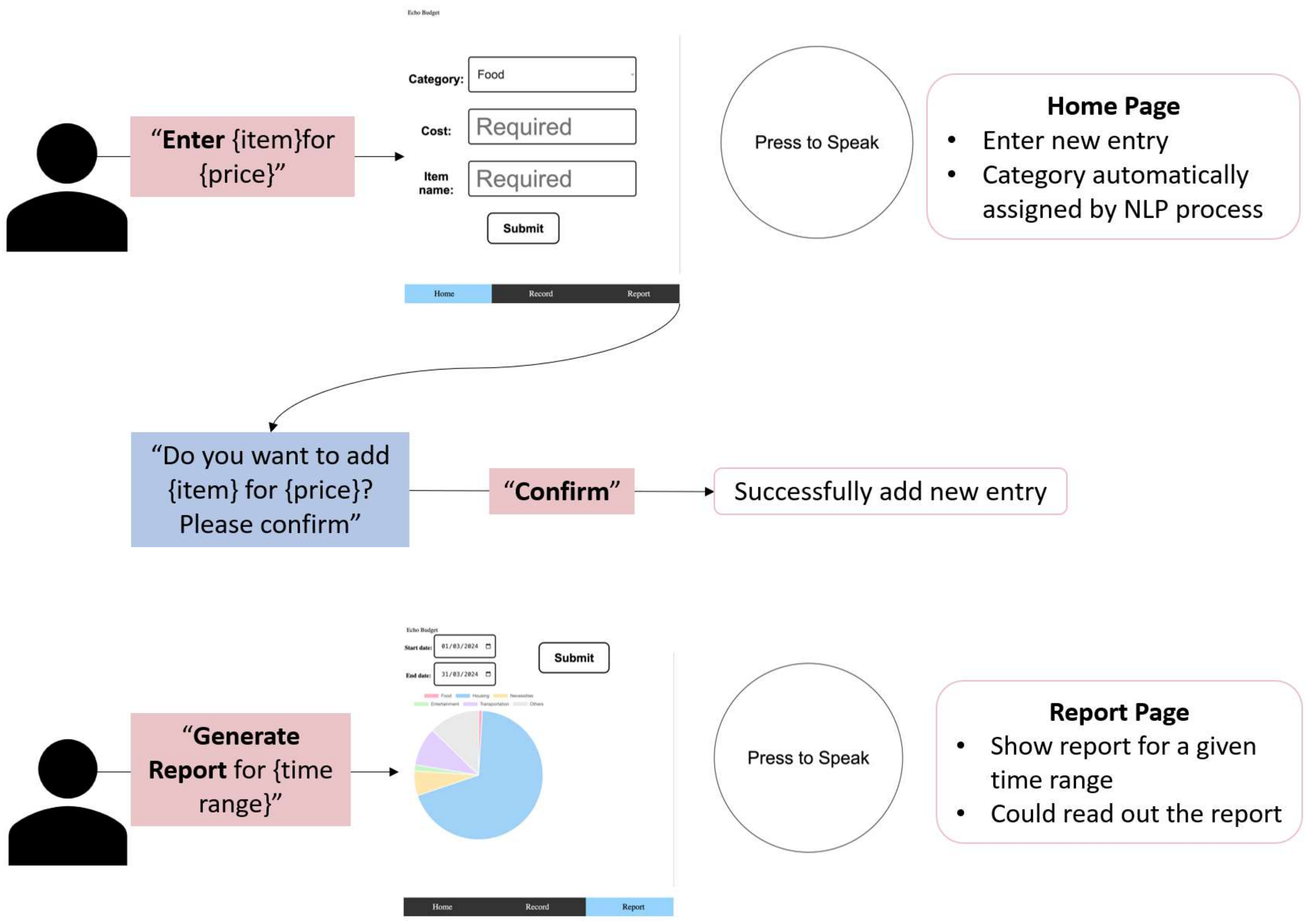
Our system performs all functionalities on a Raspberry Pi 4 with a touchscreen monitor and a USB microphone. Customers could interact with the system via finger tapping and audio inputting. After an initial tutorial that provides users with a basic introduction to the web application through the built-in speaker, the primary user interface would be displayed on the monitor screen.

Customers then could start to deliver voice command inputs by clicking the big "Press to Speak" button on the screen. The audio signals would be transmitted to the speech recognition subsystem where the voice input would be transmitted to text strings. Our natural language processing (NLP) subsystem will then parse the text string, extract the necessary information for the web app, and assign categories to items if needed.

Based on the received commands, different pages of the web application would be rendered. And the web app would complete the tasks assigned by the users.



The graph below shows how customers could interact with our application. The red boxes contain the commands given by the user and the blue boxes will be the audio feedback given by our system to the users.


"**Enter** {item}for {price}"

Echo Budget
Category: Food
Cost: Required
Item name: Required
Submit
Home Record Report

Press to Speak

Home Page
- Enter new entry
- Category automatically assigned by NLP process

"Do you want to add {item} for {price}? Please confirm"

"**Confirm**"

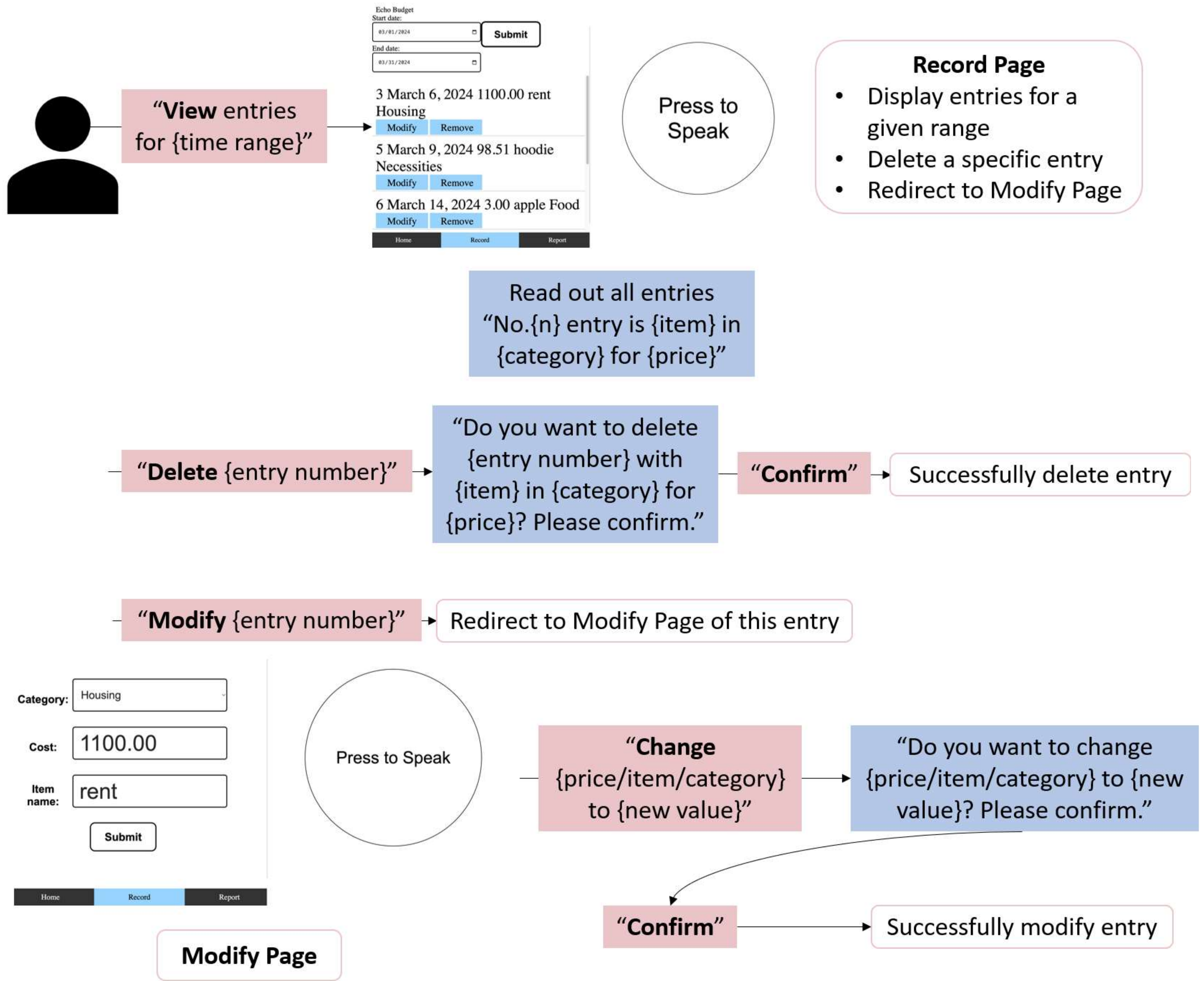
Successfully add new entry

"**Generate Report** for {time range}"

Echo Budget
Start date: 03/01/2024
End date: 03/31/2024
Submit
Home Record Report

Press to Speak

Report Page
- Show report for a given time range
- Could read out the report


"**View** entries for {time range}"

Echo Budget
Start date: 03/01/2024
End date: 03/31/2024
Submit
3 March 6, 2024 1100.00 rent Housing
Modify Remove
5 March 9, 2024 98.51 hoodie Necessities
Modify Remove
6 March 14, 2024 3.00 apple Food
Modify Remove
Home Record Report

Press to Speak

Record Page
- Display entries for a given range
- Delete a specific entry
- Redirect to Modify Page

Read out all entries "No.{n} entry is {item} in {category} for {price}"

"**Delete** {entry number}"

"Do you want to delete {entry number} with {item} in {category} for {price}? Please confirm."

"**Confirm**"

Successfully delete entry

"**Modify** {entry number}"

Redirect to Modify Page of this entry

Category: Housing
Cost: 1100.00
Item name: rent
Submit
Home Record Report

Press to Speak

"**Change** {price/item/category} to {new value}"

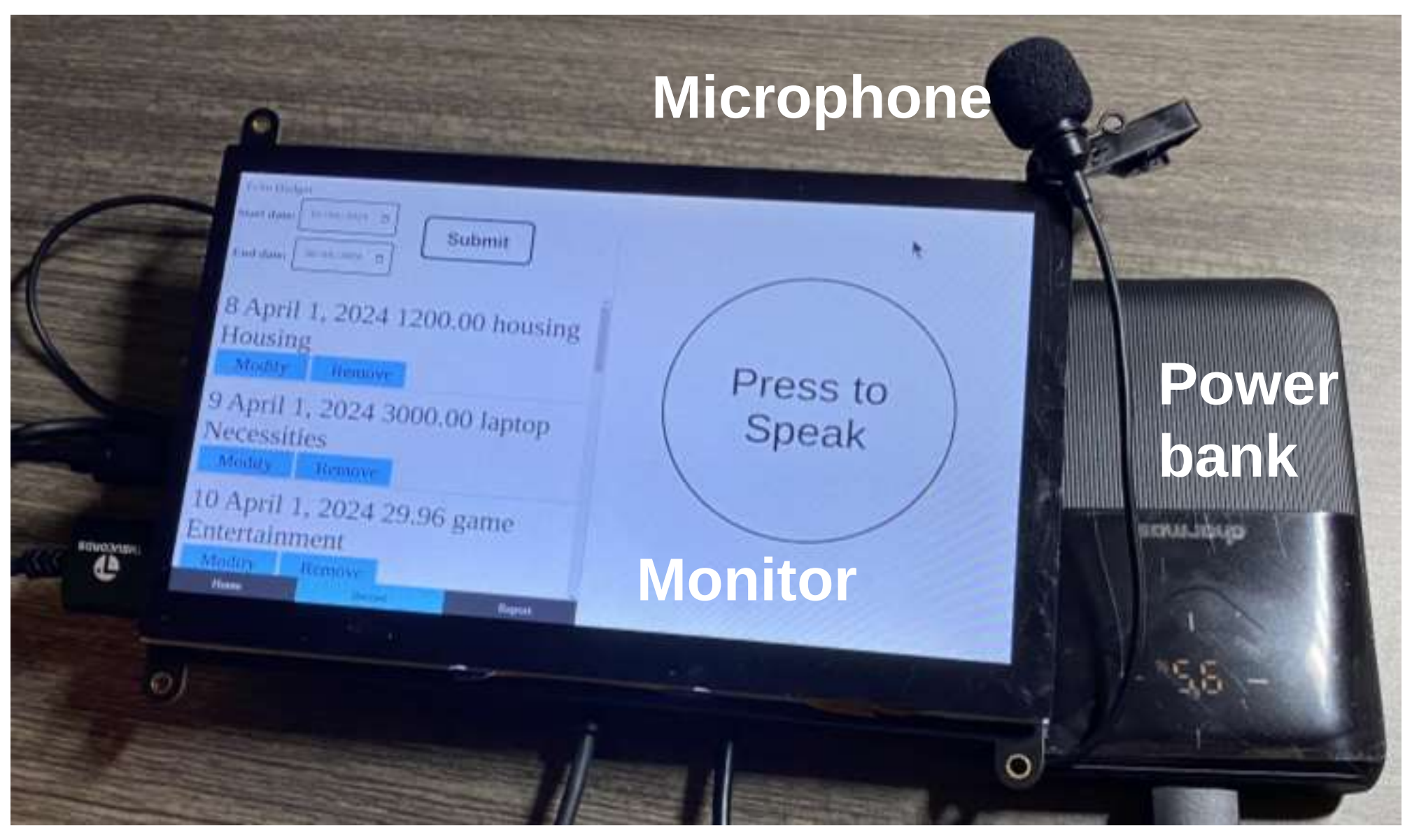
"Do you want to change {price/item/category} to {new value}? Please confirm."

"**Confirm**"

Successfully modify entry

Modify Page

System Description



System Evaluation

Test	Ideal	Actual
Latency (voice command)	4 seconds to render the page	4.52 seconds
Battery life	2 hours with monitor on	4 hours
Portability	500 grams	500 grams
Noise reduction	90% accuracy of whole system under 70dB	90%
Audio to text	Less than 20% of word error rate	98.3%
NLP parsing	Accuracy larger than 95% Takes less than 3 seconds	96% 2.5 seconds
Item classification	90% accuracy	90%
Voice response	All audio requests should be assisted with a voice response	All voice responses implemented

Design Tradeoffs	
Train one model for Spacy	**Train two models for Spacy**
Require less storage	Increase parsing accuracy Only lengthen the runtime by a little bit
Click button to stop	**Automatic speech termination detection**
More flexibility on voice input time May cause potential waste if forget to press twice	User does not need to press again Fixed voice inputting time (8 seconds)
Allow users to give commands without restriction	**Standardized voice input keywords**
More user-friendly, but hard to process	Enhance action performance accuracy
Add each item name to Glove model	**Only add item name to model when entry is modified**
More accurate, but may overfit	Reduce wait time and storage

Conclusions & Additional Information

Although our current system has good performance, we have some restrictions on user's behaviors. For example, we want users to use imperative sentences in English as the commands. One potential improvement is to allow customers to create their customized categories. Another potential improvement would be to have a better NLP model. This would allow customers to use voice commands with any sentence structure they like. We may also consider supporting other languages in the future.

Electrical & Computer ENGINEERING
Carnegie Mellon

https://course.ece.cmu.edu/~ece500/projects/s24-teamb0/