Use-Case

Money Tracker APP with Voice Input

- Customers could use EchoBudget at home and outside with the device
- Customers could verbally:
  - Record the spending
  - Change each entry
  - Request report for a given range
- **Friendly to visually impaired customers & who wants to record their expenses in a easy way**

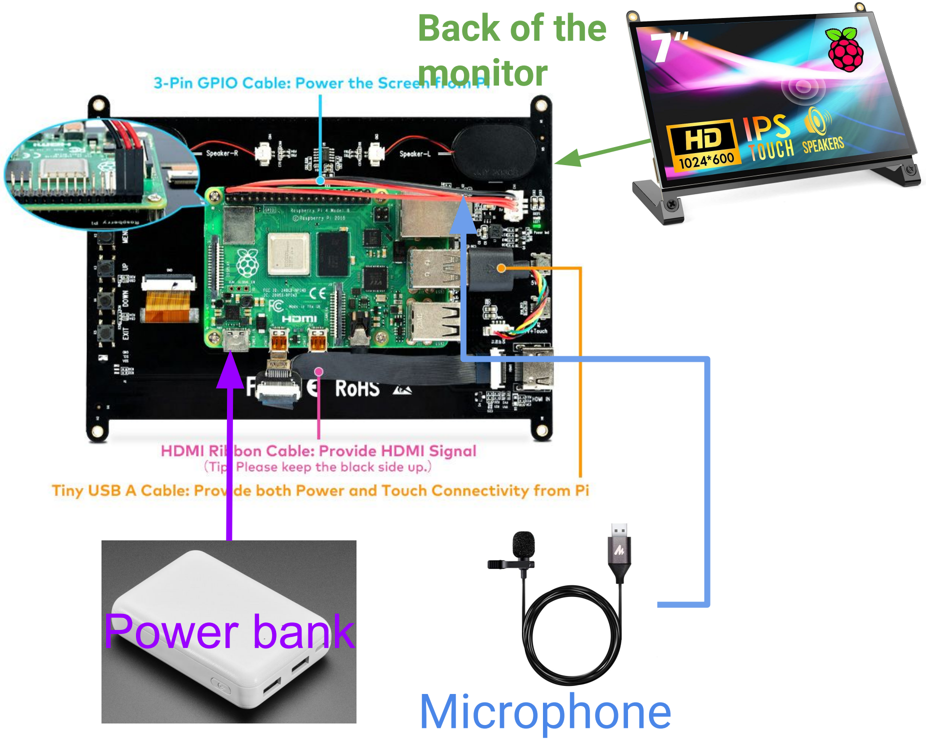
Design Requirements

- **Hardware Requirements**
  - Weight: <= 500g (weight of iPad Air)
  - 1-hour battery life (with monitor on)
- **Speech Recognition**
  - The speech to text accuracy for whole sentence >= 80%
  - The accuracy for important words >= 90%
    - Instruction (edit, generate report, etc.)
    - Amount of Spending (the number)
    - Items name

Design Requirements

- **NLP**
    - Accuracy for parse the action/money amount/items name from script >= 95%
    - Accuracy for categorization should >= 90% for our given categories
        - Food, housing, necessities, entertainment, transportation, others
        - No customized categories by MVP
- **Web App**
    - Generate Report <= 100 ms
    - Concise user interface (details in later pages)

Solution Approach



Back of the monitor

3-Pin GPIO Cable: Power the Screen from Pi

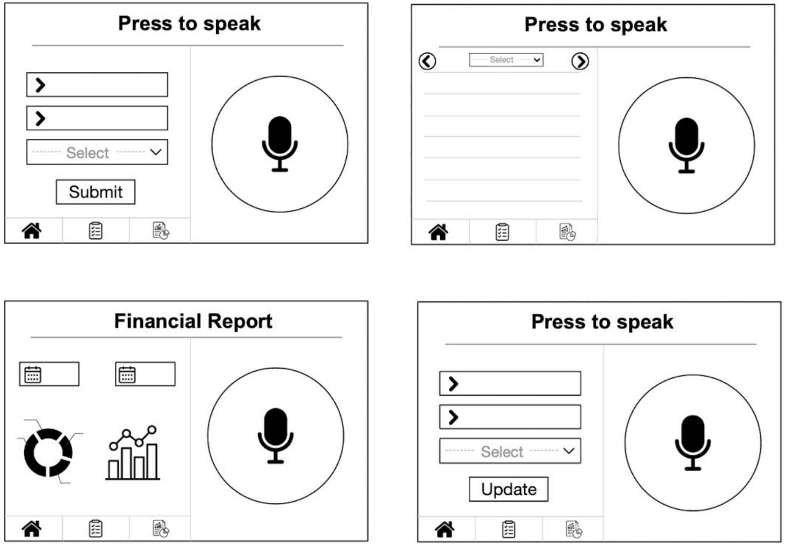
HDMI Ribbon Cable: Provide HDMI Signal
(Tip: Please keep the black side up.)

Tiny USB A Cable: Provide both Power and Touch Connectivity from Pi

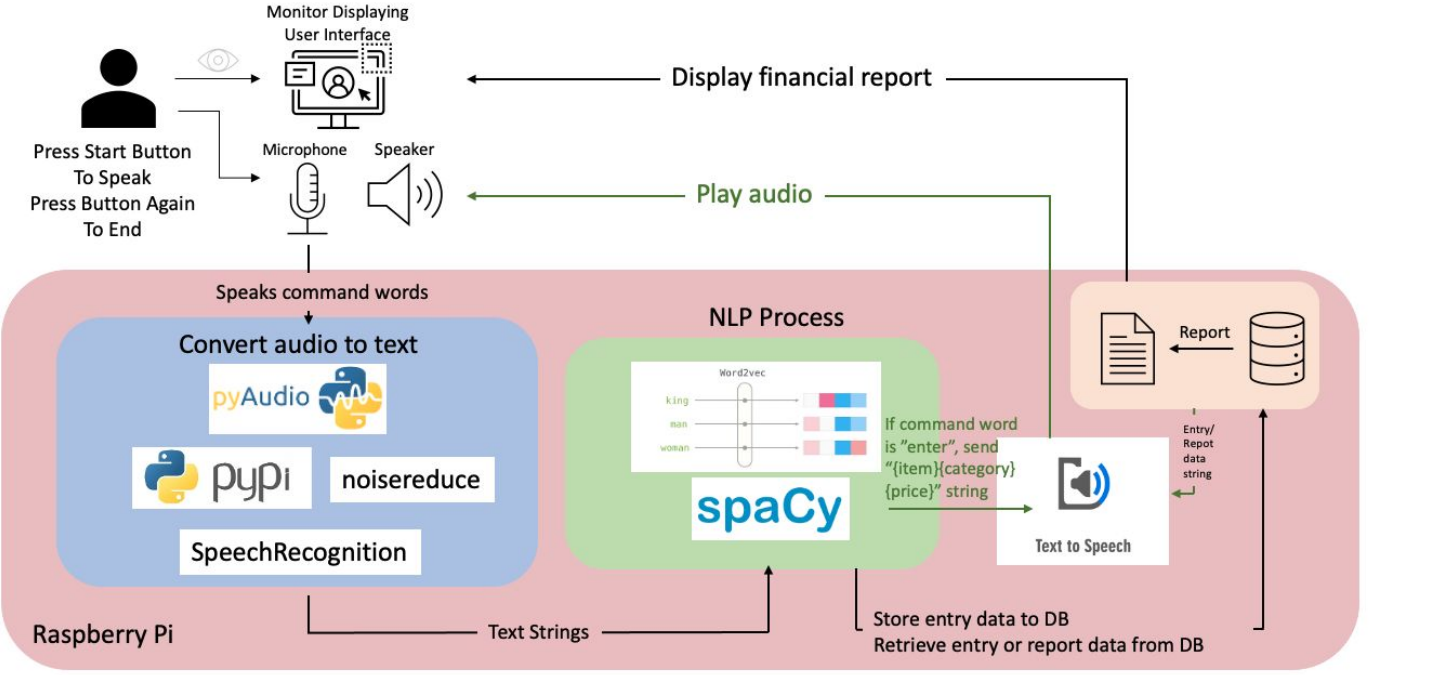
Power bank

Microphone

Web Application



Solution Approach: Block Diagram



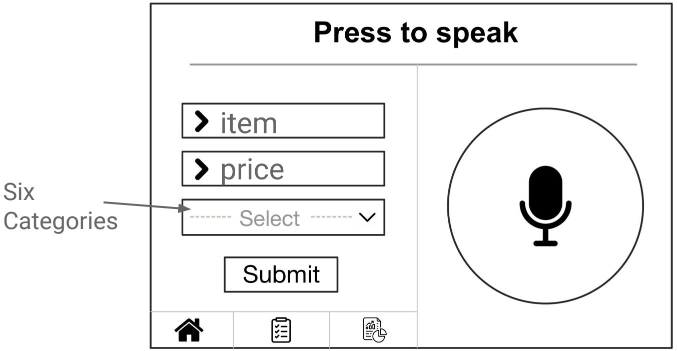
Normal Mode

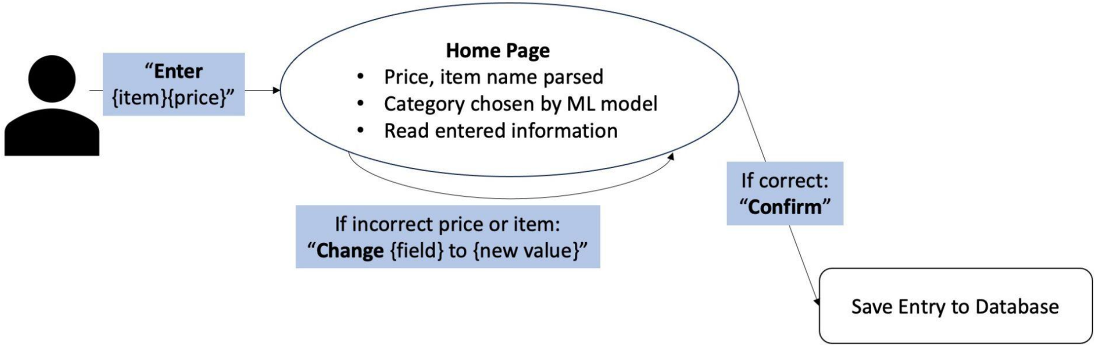
Audio Assistant Mode

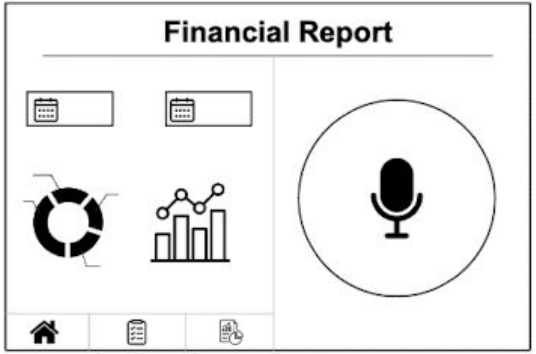
System Specification: Signal Processing

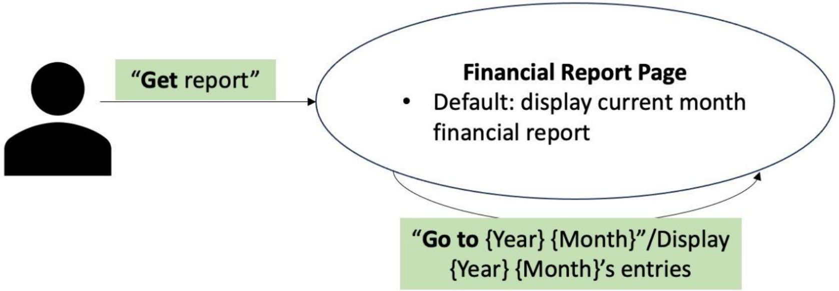
- **Speech Recognition**
  - USB microphone: record voice commands via pressing button
  - PyAudio -> PyPI noisereduce -> SpeechRecognition
    - PyAudio: Record user audio, save as .wav file
    - PyPi noisereduce: reduce noise for .wav file
    - SpeechRecognition: Convert speech to text strings
- **Natural Language Processing**
  - spaCy: command, price, and item name parsing from text string
  - word2vec: item classification
    - 6 categories: food, housing, necessities, entertainment, transportation, others
    - GloVe: Word Embedding

System Specification: Software

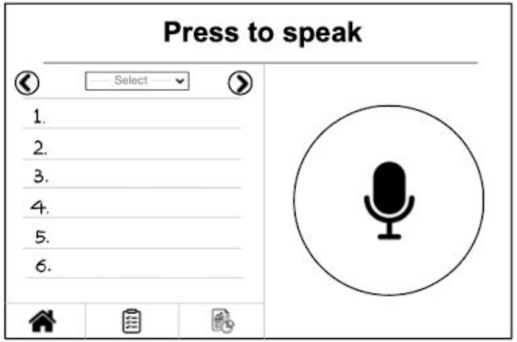


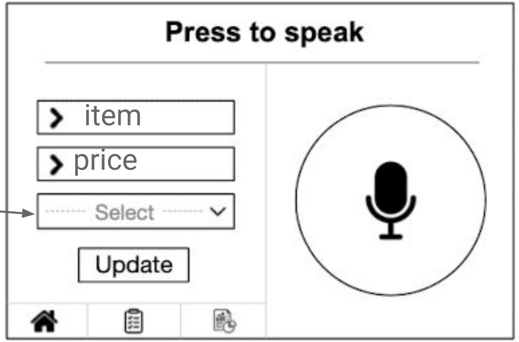

"Enter {item}{price}"
Home Page
• Price, item name parsed
• Category chosen by ML model
• Read entered information
If incorrect price or item: "Change {field} to {new value}"
If correct: "Confirm"
Save Entry to Database



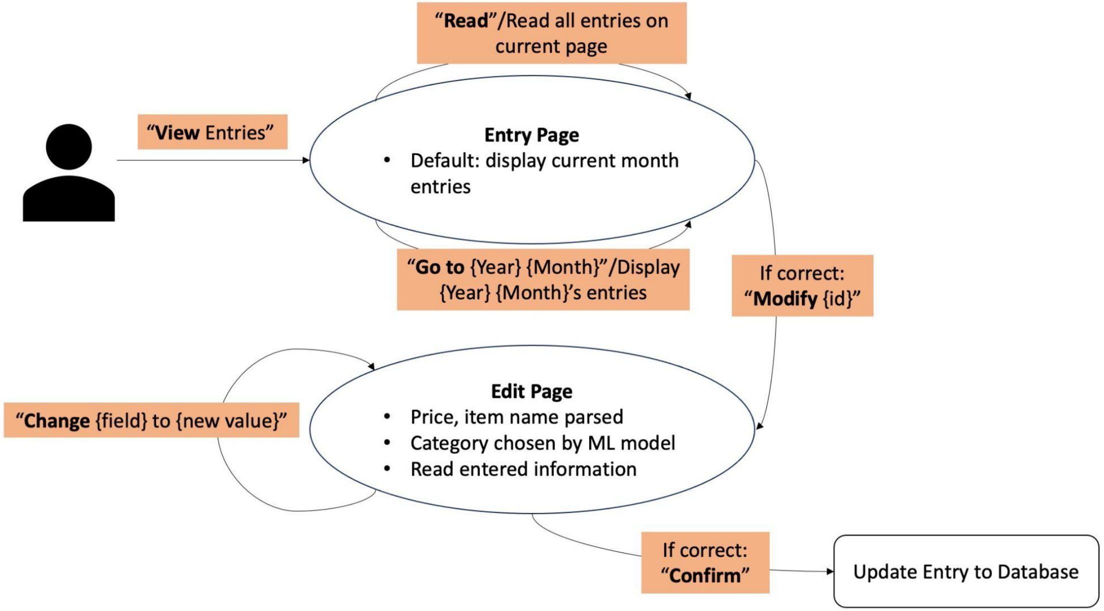

"Get report"
Financial Report Page
• Default: display current month financial report
"Go to {Year} {Month}"/Display {Year} {Month}'s entries

System Specification: Software





Six Categories


"View Entries"
"Read"/Read all entries on current page
Entry Page
Default: display current month entries
"Go to {Year} {Month}"/Display {Year} {Month}'s entries
If correct: "Modify {id}"
"Change {field} to {new value}"
Edit Page
Price, item name parsed
Category chosen by ML model
Read entered information
If correct: "Confirm"
Update Entry to Database

Implementation Plan

- **Software**
  - Developing: Django framework Web Application
  - Downloading: Speech Recognition models
    - PyAudio, PyPI noisereduce, SpeechRecognition
  - Modifying/Training: Natural Language Processing models
    - spaCy, word2vec
  - Text-To-Speech model: gTTS
  - Database: Django’s SQLite database
- **Hardware**
  - Raspberry Pi and monitor with speaker
  - USB microphone

Testing, Verification and Metrics

Performance Tests

Test	Input	Output	Expected Performance
Speech recognition	Digital signal of audio input	Text (list of words)	Less than 20% of word error rate (WER)
Text to command (NLP)	Text output from above	Command (verb + params)	Identify verb with 100% accuracy; Identify item name and money with 95% accuracy; classify category with 90% accuracy; NLP process takes less than 3s
Financial report	Date	Report	Generated within 100 ms
Battery life	NA	NA	Should last for 1 hour with the monitor on

Testing, Verification and Metrics

Functional Tests

Make sure the app works as defined in the requirements. Tests will be specified in design report in the following format:

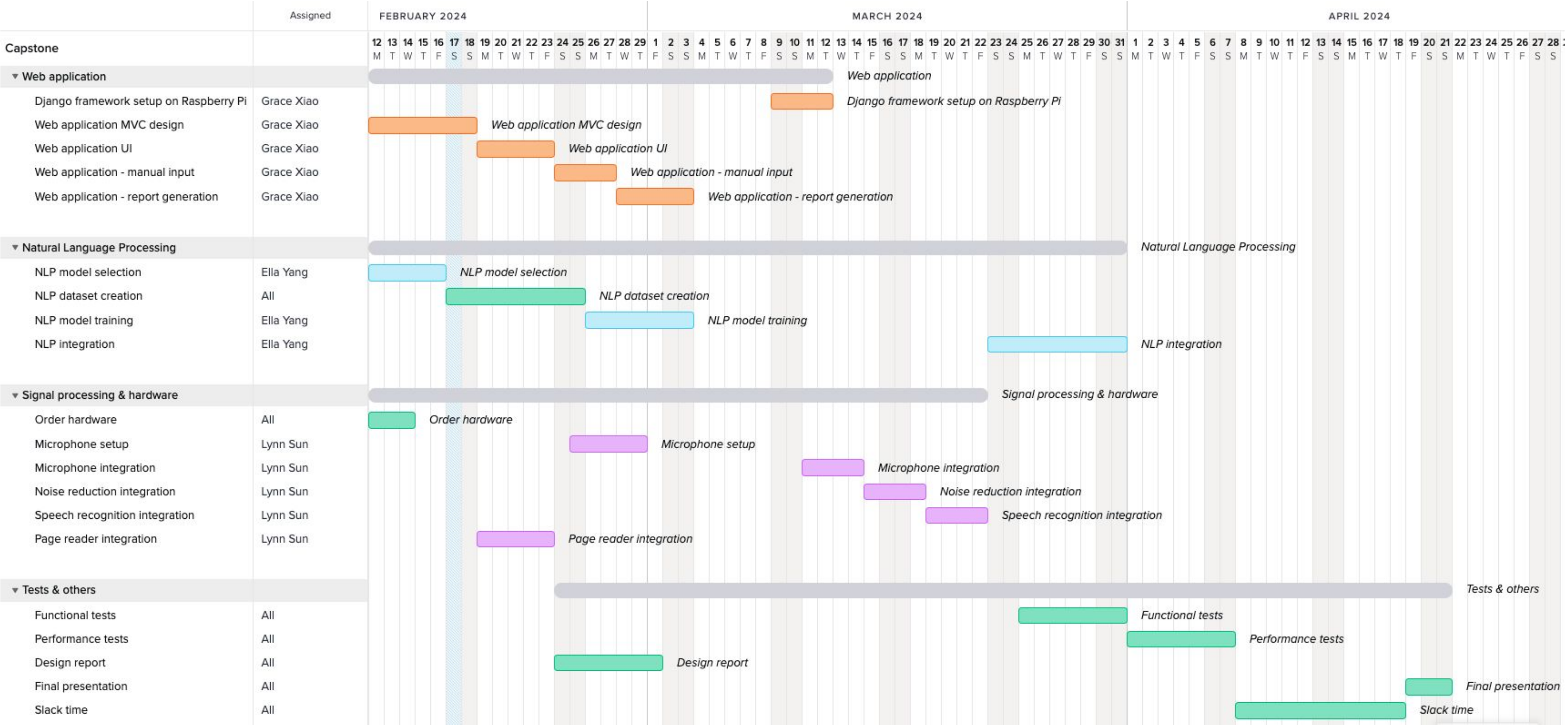
Start page/state, operation, resulting page/state

Risk Mitigation

Conduct operations manually if low accuracy or speed

Create an instruction page with recognizable commands

Schedule



Capstone	Assigned
▾ Web application	
Django framework setup on Raspberry Pi	Grace Xiao
Web application MVC design	Grace Xiao
Web application UI	Grace Xiao
Web application - manual input	Grace Xiao
Web application - report generation	Grace Xiao
▾ Natural Language Processing	
NLP model selection	Ella Yang
NLP dataset creation	All
NLP model training	Ella Yang
NLP integration	Ella Yang
▾ Signal processing & hardware	
Order hardware	All
Microphone setup	Lynn Sun
Microphone integration	Lynn Sun
Noise reduction integration	Lynn Sun
Speech recognition integration	Lynn Sun
Page reader integration	Lynn Sun
▾ Tests & others	
Functional tests	All
Performance tests	All
Design report	All
Final presentation	All
Slack time	All