Team C0: Backpack Buddy

A smart inventory system for on-the-go students

Team Members: Joon Cha, Aaron Li, Janet Li

Presented by: Aaron Li

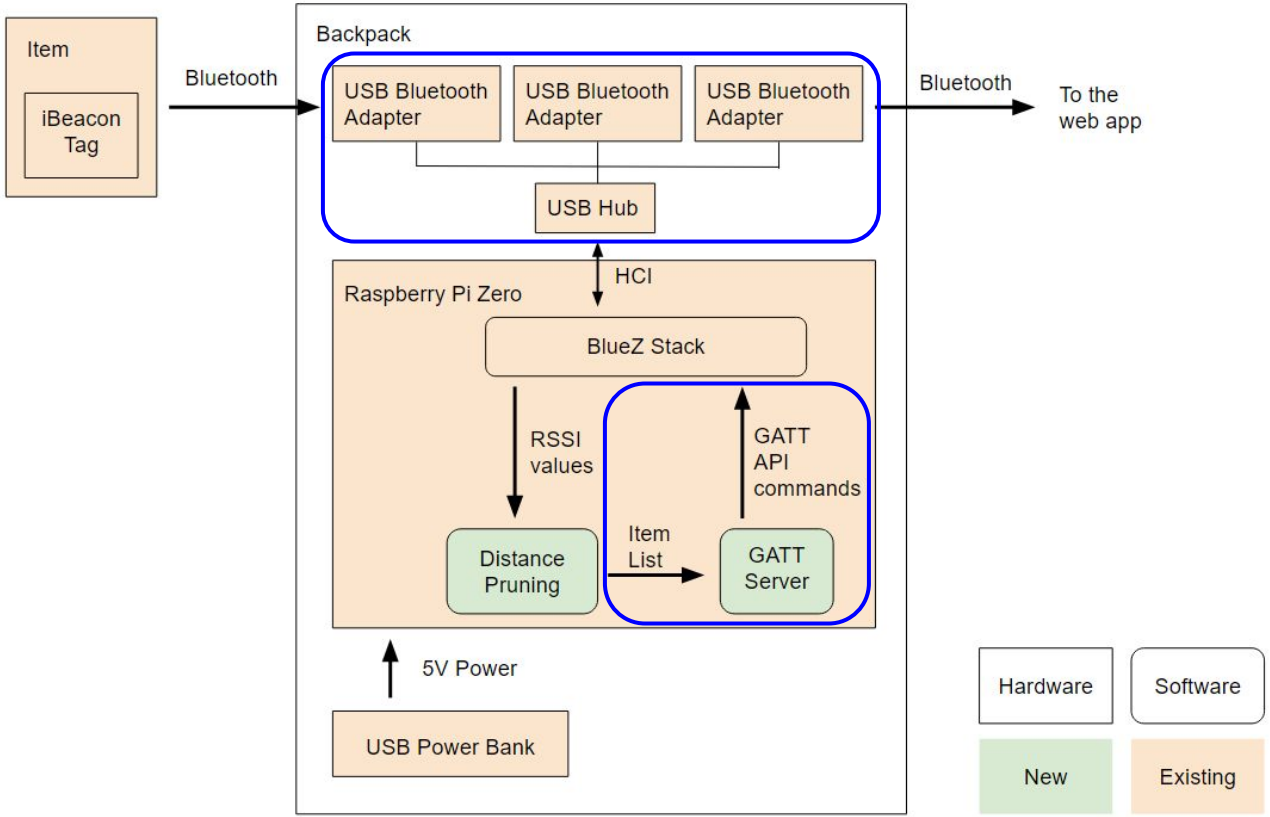
Application Area

- Keeping track of items is hard
- Other tags only focus on locating individual items
- **Backpack Buddy helps students manage groups of items in relation to their schedule**
  - Tag items and assign them to your events
  - Get a phone notification when you are missing items
  - Also get notified if you leave an item behind
- Optionally, Backpack Buddy can also **learn your schedule** by analyzing what items you bring when and where
- Main requirements:
  - Detect what tags are in the backpack
  - Display item list to the user and report missing items

Solution Approach: Hardware

- Bluetooth Low Energy tags
  - Attached to each item with a keyring or double-sided tape
- Raspberry Pi Zero
  - Small + low power draw (lasts 18+ hours on USB power bank)
- 3 USB Bluetooth Adapters
  - Additional sensing capability at a low cost
  - Simple integration using USB hub + extension cables
- Bluetooth GATT Server
  - Broadcasts a Bluetooth signal
  - Once connected, reports item list once per second

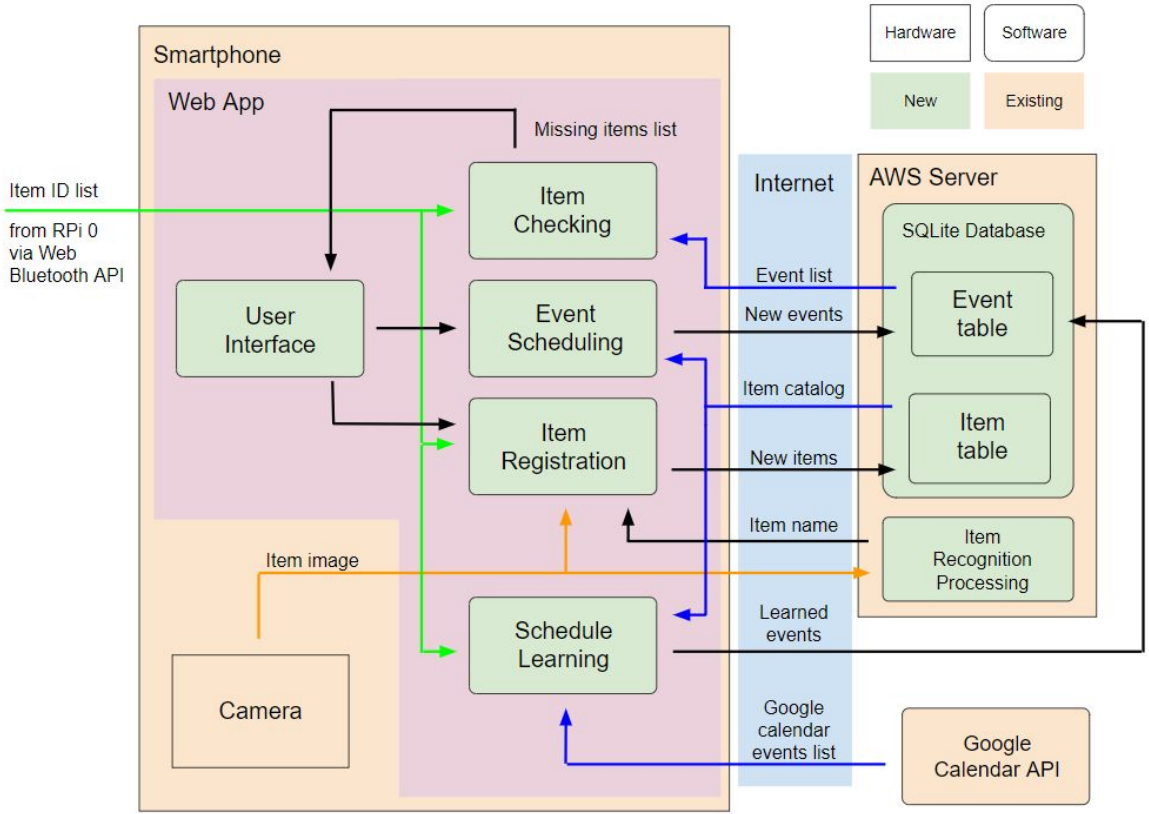
Solution Approach: Hardware Diagram



Solution Approach: Software & Signals

- Registration
  - Suggest item name using computer vision item recognition
    - CNN model for the item recognition using VGG16 in Tensorflow/Keras
- Scheduling
  - Database storage and retrieval
  - Associate items with every event
- Item checklist and notification
  - Users can specify when to notify them of missing items
  - Also notifies of lost (left behind) items after event
- Schedule learning
  - Import user's calendar events using GCal API
  - System learns which items users bring to which events

Solution Approach: Software Diagram



Complete Solution

Trade-offs

- Update speed vs. item list accuracy
  - We chose to lower the update frequency in exchange for better item detection accuracy
  - Knowing what items are in the backpack **accurately** > **quickly**
- Bluetooth Low Energy (BLE) vs. RFID
  - Important considerations: budget constraints, ease of integration with other components
  - RFID tags cheaper, but readers > $800-$1600
  - BLE allows for easy integration with other system components; RFID does not
- Convolutional Neural Network (CNN) vs. other ML models
  - CNN's automatically extract features from images (no feature engineering required)
  - CNN's scalable for large datasets
- Web app vs Native Android App
  - Web apps are more universal than a native Android app
  - Native Android app is slower and more difficult to develop
    - Harder to test/deploy

Requirements

Requirement	Test Method	Expected Result
< **1s** delay for tag detection	Place item into backpack and check server output	Item list updates within 1s
< **3s** delay for interface update	Place item into backpack and check phone screen	Webpage updates within 3s
Detect **10** items within a 0.5-meter range ± 0.1m with **100%** accuracy	Place 10 items placed into backpack	Item list displays all 10 correct ID's
Last **18 hours** w/o recharge	Insert/remove one item every 30 minutes from the backpack	System still on after 18 hours
Handle **25** weekly events	Create 25 one-hour long events in the app spread out across the week	All events appear in calendar
Notification appears 5 minutes before event which indicates **exactly** which items are missing	All items for event placed in backpack except one or two randomly selected items	Notification with correct list of missing items appears on phone screen
80% item recognition accuracy	Input images from dataset of common items	Correct name of item 80% of the time

Results

Requirement	Expected Result	Results	
< **1s** delay for tag detection	Item list updates within 1s	❌	Updated between 3.86s to 8.19s
< **3s** delay for interface update	Webpage updates within 3s	✔	Webpage updates < 1s after system update
Detect **10** items within a 0.5-meter range ± 0.1m with **100%** accuracy	Item list displays all 10 correct ID's	✔	10 tags were attached to various objects (some made of metal) and all 10 ID's showed up
Last **18 hours** w/o recharge	System still on after 18 hours	✔	System lasted 18+ hours on 5 trials
Handle **25** weekly events	All events appear in calendar	✔	Web app successfully stored and displayed 25 events
Notification appears at specific time before event which indicates **exactly** which items are missing	Notification with correct list of missing items appears on phone screen	✔	Notification appeared on desktop in Windows and also appeared in Android
80% item recognition accuracy	Correct name of item 80% of the time	✔	84.36% with total 3150 test image datasets (150 images per student items)

Project Management

Aaron	**Janet**	**Joon**
• RPi scanner and tag setup • Distance pruning algorithm • Bluetooth communication protocol • AWS deployment	• Event scheduling (web app) • Item registration (web app) • Schedule learning	• Item recognition CV algorithm • Item registration • Integration with web app

What changed:	**Work to be completed before final demo:**
• Using a Web app instead of a native Android app • No longer implementing sleep protocol for RPi Zero (to conserve battery)	• Distance pruning algorithm refinement • Schedule learning • Bluetooth persistence (thin client) • Integration with image recognition component • Compatibility with Google Calendar

Project Management

All
Joon
Janet
Aaron
Janet & Aaron
Joon & Janet

Backpack Buddy Timeline

Total Slack: 2 weeks (15 days)

TASK NAME	TEAM MEMBER	PERCENT COMPLETE
Documentation		
Abstract	All	100%
Proposal Presentation	All	100%
Design Review Presentation	All	100%
Design Review Report	All	100%
Final Presentation	All	80%
Product Filming & Video Editing	Aaron	0%
Poster	All	0%
Final Report	All	0%
App		
Design user flow diagram for phone app	All	100%
Wireframe UI's	Janet	100%
Set up database	Janet	100%
Event creation/assignment	Janet	100%
Tagged item checklist	Janet & Aaron	100%
Deletion of items/events	Janet	100%
Event notifications	Janet	100%
Missing item notifications	Janet & Aaron	100%
Integrate with camera registration	Joon & Janet	0%
Make Bluetooth connection persistent	Janet & Aaron	0%
Deployment	Aaron	100%
Backpack Device + Tags		
Purchase tags	All	100%
Design system	All	100%
Communication with tags	Aaron	100%
Add battery to RPi Zero	Aaron	100%
Communication with web app	Janet & Aaron	100%
Incorporate additional Bluetooth sensors	Aaron	100%
Mount RPi to backpack	Aaron	100%
Automate server startup	Aaron	100%
Investigate reducing pairing time	Aaron	0%
Camera Registration		
Research existing item recognition work	All	100%
Determine items to be recognized	All	100%
Design process	Joon	100%
Learn algorithm to recognize items	Joon	100%
Research CNN models for the item recognition	Joon	100%
Collect student item image and augment images	Joon	100%
Implement the CNN model	Joon	100%
Train the CNN model	Joon	100%
Finalize the CNN model	Joon	100%
Set up endpoint for web app	Joon	0%
Testing (and debugging)		
Tag detection testing	Aaron	100%
Backpack physical movement testing	Aaron	100%
Scanner battery testing	Aaron	100%
Interface display delay testing	Janet & Aaron	100%
Event testing	Janet	100%
Item recognition testing	Joon	100%
Usability testing	Janet	100%
Distance pruning algorithm testing	Aaron	0%
Schedule Learning		
Collect sample data	Janet	100%
Test event occurrences against collected data	Janet	0%
Formally assign items to events	Janet	0%
Slack Time		
Slack Time	All	0%