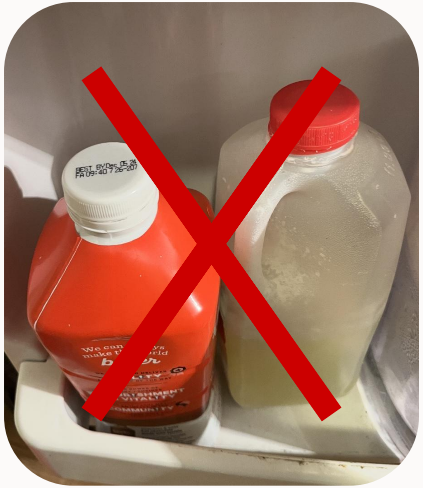
GlowFresh

Silicone Pad for Reducing Food Waste Through Expiration Tracking

Team F2 - **Jessica Chan**, Gina Seo, Sarah Yang



103 lbs

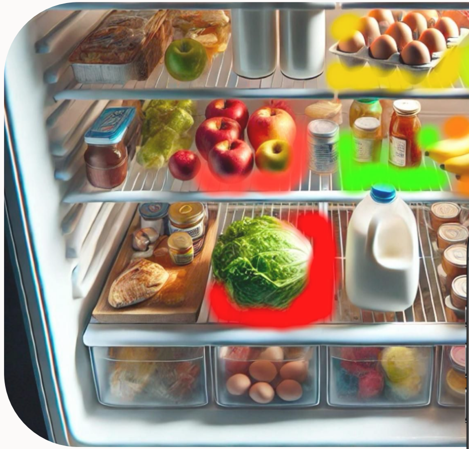
wasted per person every year
from food expiring in the fridge

GlowFresh makes food freshness visible, so you never forget and waste.

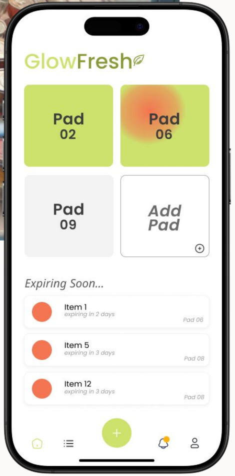
- A silicone pad that glows to indicate food freshness with accurate expiration predictions

- Paired with a mobile app, it allows users to register perishables and suggests smarter grocery choices to prevent spoilage

Hardware Software



2 x 2 pad



GlowFresh

Pad 02

Pad 06

Pad 09

Add Pad

Expiring Soon...

Item 1 expiring in 2 days Pad 06

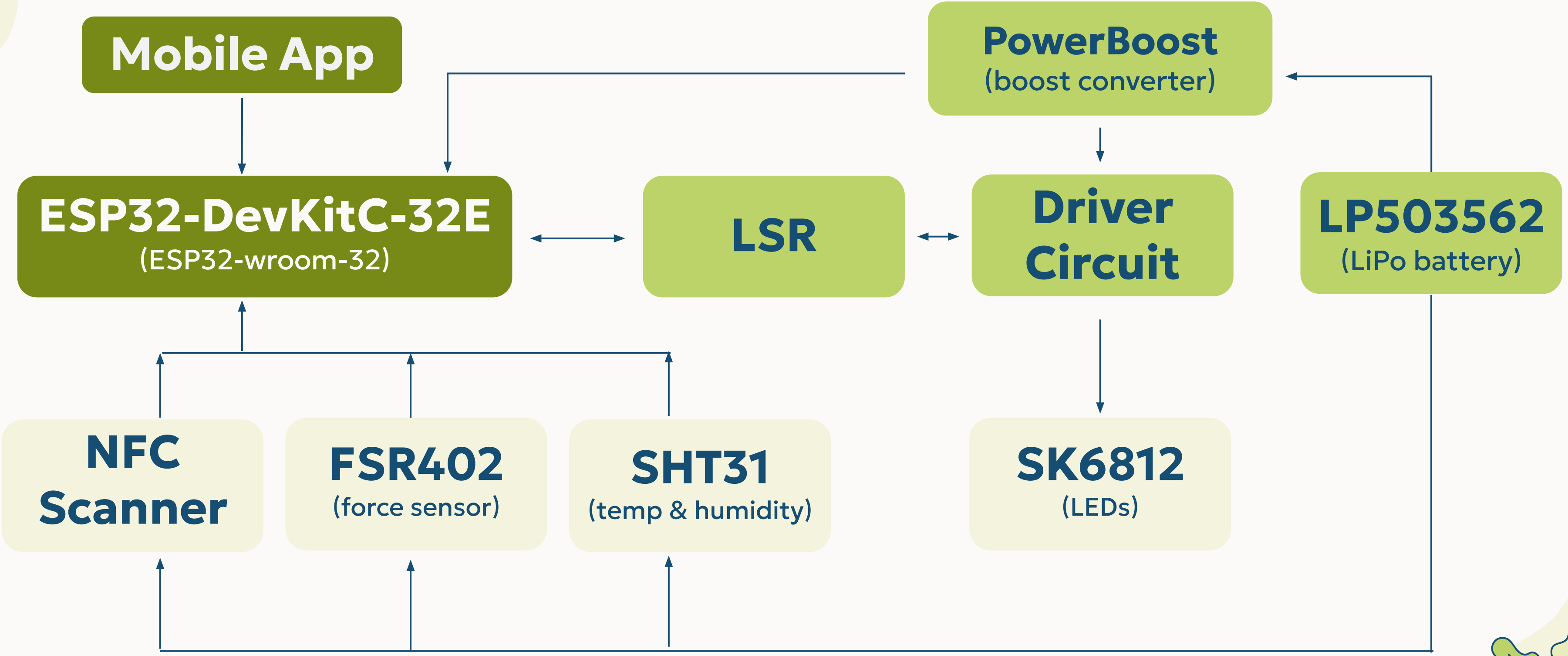
Item 5 expiring in 3 days Pad 08

Item 12 expiring in 3 days Pad 08

Requirements & Challenges

Use case	Design Requirements
Should accurately detect an item placed on the pad and display expiration	**Pressure:** Detects ≥ 50g, supports 2 stacked items. **Detection:** ≥ 90% accuracy, 90/100 tests to recognize placement. **LEDs:** Clear transitions & synced with date (Green → Yellow → Red) **Multi-Item Tracking:** Handles 8 items/pad, ≤ 3s sync delay
Users should be notified of expiration and warned beforehand	**Notifications**: 100% success rate, 2 reminders before red
Product itself should be easy to use, low maintenance and food safe	**Battery Life**: ≥ 1 month **Food-Safe Materials**: FDA-approved silicone. **User-Friendly**: ≤ 3 steps to log an item; manual item removal; clearly indicates which item in stack is expiring, intuitive UI

System Solution (Hardware)



Mobile App

PowerBoost
(boost converter)

ESP32-DevKitC-32E
(ESP32-wroom-32)

LSR

Driver Circuit

LP503562
(LiPo battery)

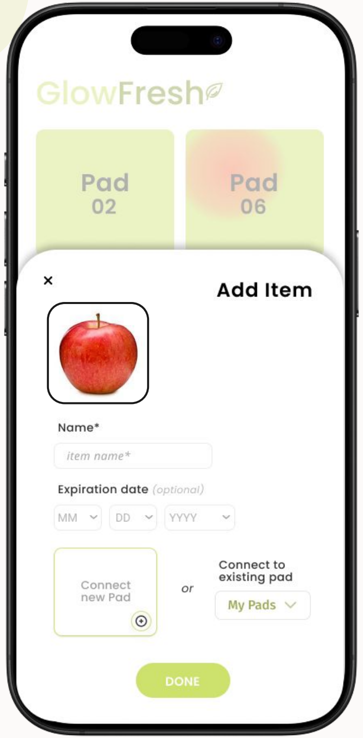
NFC Scanner

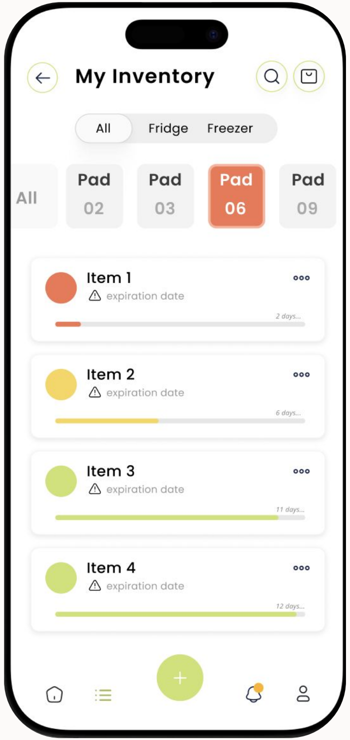
FSR402
(force sensor)

SHT31
(temp & humidity)

SK6812
(LEDs)

System Solution (Software)



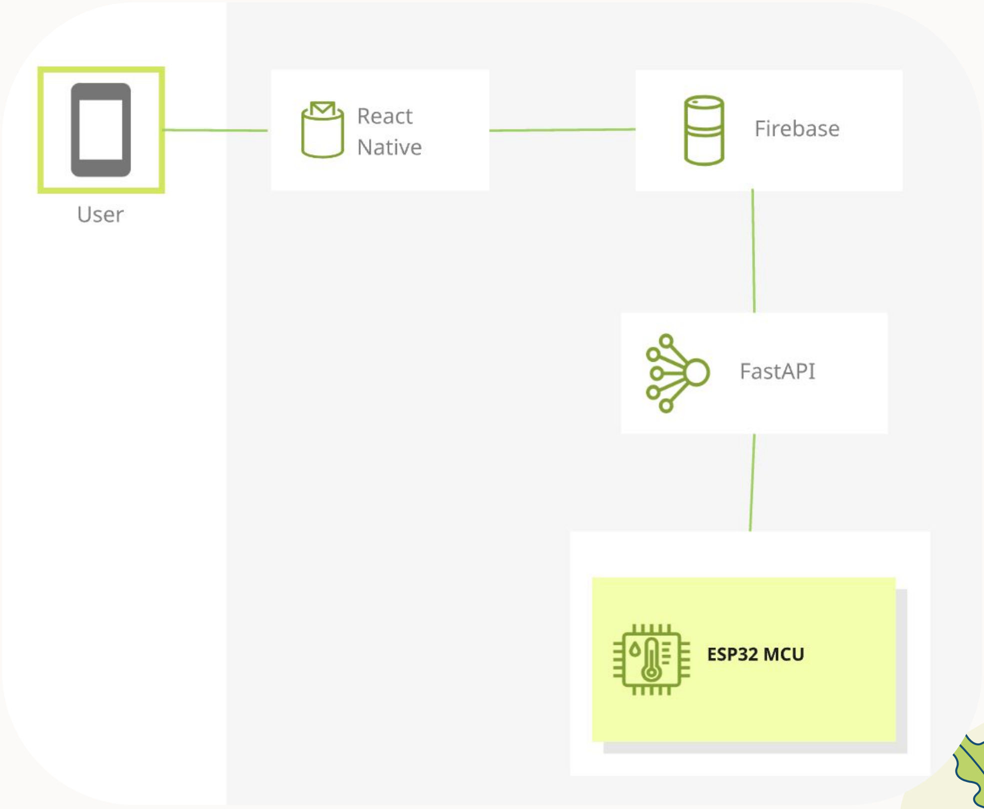


Item 1
expiration date
2 days...

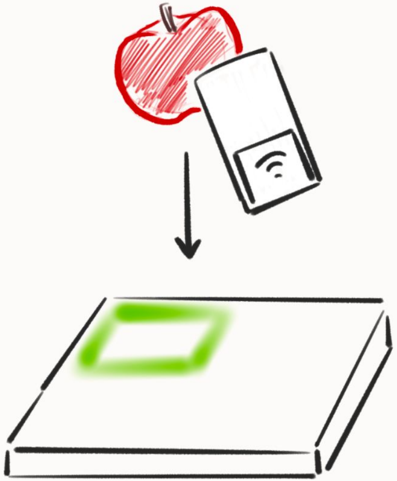
Item 2
expiration date
6 days...

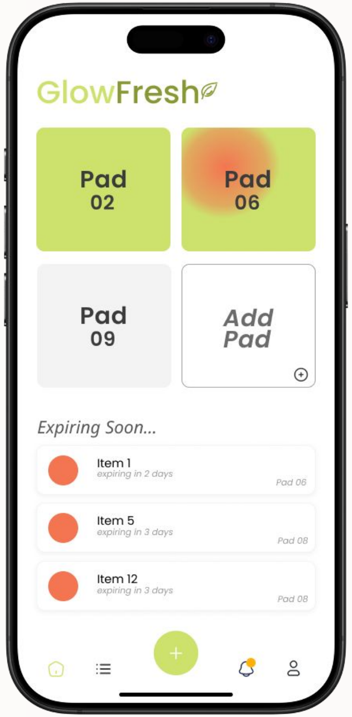
Item 3
expiration date
11 days...

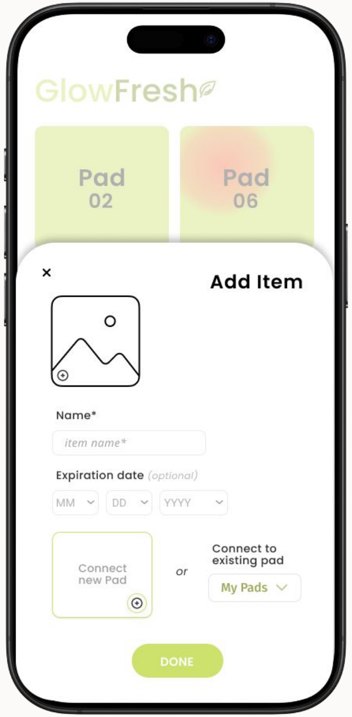
Item 4
expiration date
12 days...

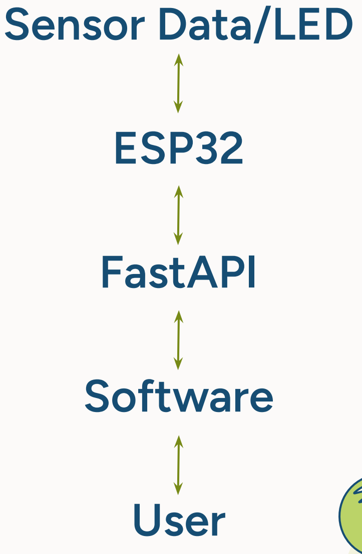


Complete Solution









Sensor Data/LED

ESP32

FastAPI

Software

User

Verification Tests

Feature	Requirement	Input	Method
Pressure detection	> 98% accuracy	Items 50g - 200g	Place varying weights (50g, 100g, 150g, etc)
Expiration to LED update	< 10s	Non-expiring, warning, expired times	Preload sample entries
NFC scan	< 5s	NTAG215 stickers	Scanned multiple tags and measured time to detect item
Item location detection	99% accuracy	Various NFC stickers on multiple items	Logged items in multiple zones, moved them around

Validation Tests

Feature	Requirement	Input	Method
Manual add item -> inventory update	<= 3s, no missed items	No NFC tag, manual input	Item added, ensure firebase + UI updated
Scan add item -> inventory update	<= 5s, no missed detections	Food item with NFC sticker	Scan tag, ensure UID sent to firebase + inventory updated with correct info
Item removal -> inventory update	<= 3s	1. Moving item to different zone 2. Deleting item	Item removal from certain zone, deleting item completely from inventory
Expiring item -> Notification	99% accuracy	Items in warning and expired timeframe	Preload items of varying expiration

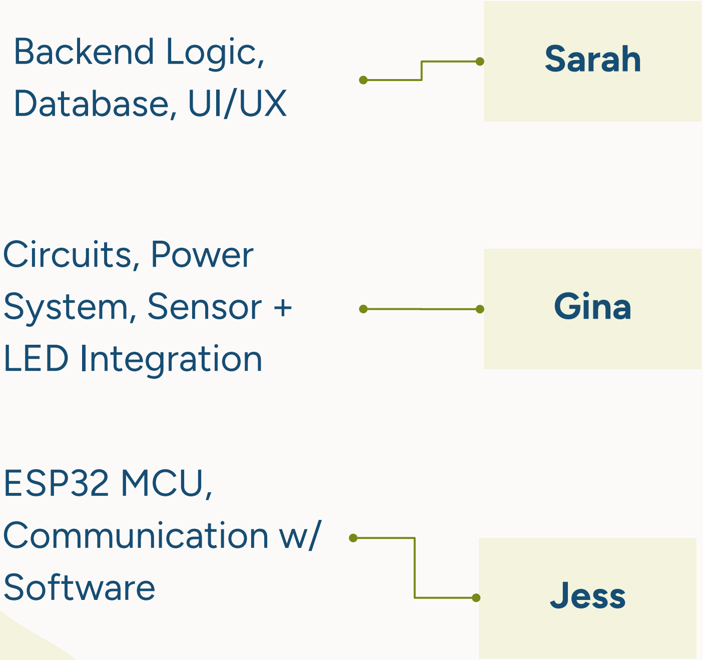
Results

Feature	Target	Result	Notes
Expiration -> LED update	< 10s	< 10s	FastAPI sends periodic updates to ESP32, we set time to 10s to reduce overhead as items usually expire on a daily basis
Item location detection	99% accuracy	100%, < 10s update time	ESP32 sends sensor data updates every 10s
Manual add inventory update	<= 3s	2s average, no missed items	
Scan add inventory update	<= 5s	4s average, no missed detections	Phone scanner takes < 2s, board scanner takes longer ~5s
Item removal update	<= 3s	<2s	
Expiring item notification	99% accuracy	100% correct notifications	Sends notifs on change in expiration status

Design Trade-offs

Original Plan	Changes	Trade-off
Thin, polished PCB	Breadboard for modular prototyping	Bulkier (~150% thicker), but saved ~1 week for iteration and improved reliability
Fully molded silicone casing	Modular silicone encasing	Less polished look, but allowed easy hardware swaps
NFC scanner underneath pressure sensors	Side-by-side sensor layout	Sacrificed compactness to ensure reliable NFC scanning
3 registration paths (App + Mat)	Mat-only fallback if App NFC fails	Lost 1 method, preserved core functionality

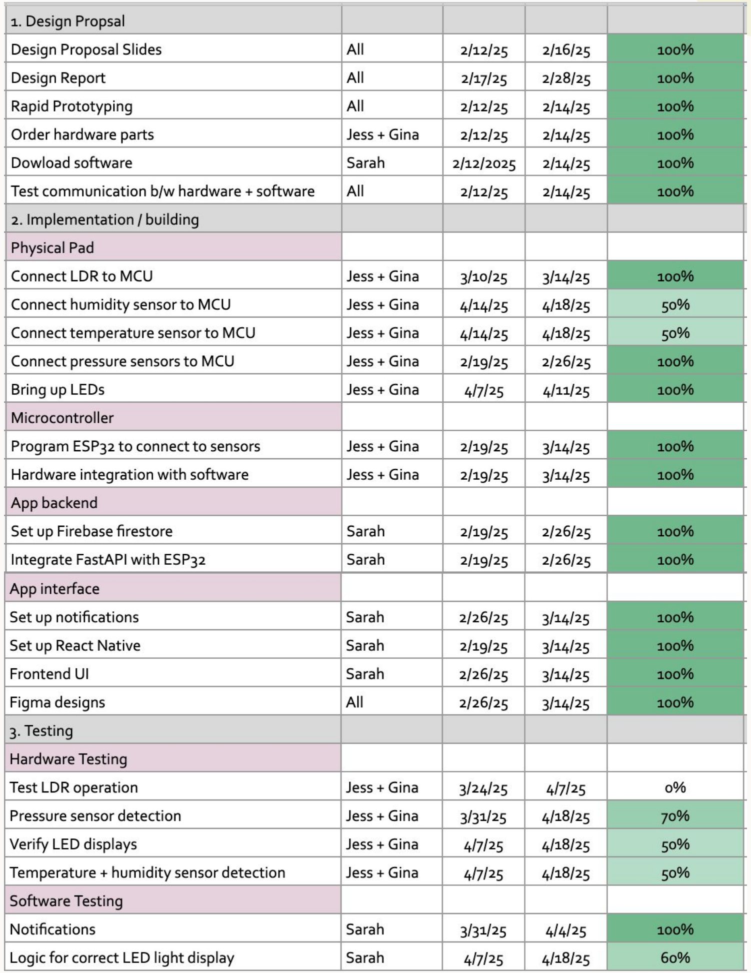
Project Management



Backend Logic, Database, UI/UX → Sarah

Circuits, Power System, Sensor + LED Integration → Gina

ESP32 MCU, Communication w/ Software → Jess



1. Design Propsal				
Design Proposal Slides	All	2/12/25	2/16/25	100%
Design Report	All	2/17/25	2/28/25	100%
Rapid Prototyping	All	2/12/25	2/14/25	100%
Order hardware parts	Jess + Gina	2/12/25	2/14/25	100%
Dowload software	Sarah	2/12/2025	2/14/25	100%
Test communication b/w hardware + software	All	2/12/25	2/14/25	100%
2. Implementation / building				
Physical Pad				
Connect LDR to MCU	Jess + Gina	3/10/25	3/14/25	100%
Connect humidity sensor to MCU	Jess + Gina	4/14/25	4/18/25	50%
Connect temperature sensor to MCU	Jess + Gina	4/14/25	4/18/25	50%
Connect pressure sensors to MCU	Jess + Gina	2/19/25	2/26/25	100%
Bring up LEDs	Jess + Gina	4/7/25	4/11/25	100%
Microcontroller				
Program ESP32 to connect to sensors	Jess + Gina	2/19/25	3/14/25	100%
Hardware integration with software	Jess + Gina	2/19/25	3/14/25	100%
App backend				
Set up Firebase firestore	Sarah	2/19/25	2/26/25	100%
Integrate FastAPI with ESP32	Sarah	2/19/25	2/26/25	100%
App interface				
Set up notifications	Sarah	2/26/25	3/14/25	100%
Set up React Native	Sarah	2/19/25	3/14/25	100%
Frontend UI	Sarah	2/26/25	3/14/25	100%
Figma designs	All	2/26/25	3/14/25	100%
3. Testing				
Hardware Testing				
Test LDR operation	Jess + Gina	3/24/25	4/7/25	0%
Pressure sensor detection	Jess + Gina	3/31/25	4/18/25	70%
Verify LED displays	Jess + Gina	4/7/25	4/18/25	50%
Temperature + humidity sensor detection	Jess + Gina	4/7/25	4/18/25	50%
Software Testing				
Notifications	Sarah	3/31/25	4/4/25	100%
Logic for correct LED light display	Sarah	4/7/25	4/18/25	60%