Yuqing Ye

SELECTED WORK

ROLE
Electrical & Computer Eng.

FOCUS
Edge AI · Embedded Systems

LAB
ARC Lab — Drone Vision

EDITION
2026

01 ABOUT ME

I'm an electrical and computer engineer who starts with questions, not resources.

In freshman year, I saw students drowning in fragmented event information. I assembled a 9-person team, ran two rounds of user testing, and shipped a solution to 100+ active users. That taught me how to move fast from problem to prototype by translating messy, real-world needs into testable hypotheses and shipped artifacts.

At Halogen.AI, I built AI integrations for small business owners across industries including retail, journalism, and hospitality. Working closely with users showed me how emerging technologies can transform challenges that once seemed too costly, complex, or time-consuming to solve.

Today, I am exploring intelligent systems across the full stack. At ARC Lab, I work on deploying computer vision models for resource-constrained drone platforms, balancing accuracy, latency, power, and hardware limitations. In parallel, I am developing an intelligent robotic pet that integrates an ESP32, cloud-based language and speech services, embedded audio, motion control, and interactive hardware within a compact, low-cost system.

I am not motivated by technology for its own sake. I am motivated by its ability to solve problems that people have learned to tolerate. Whether I am building a campus platform, developing AI tools for small businesses, deploying edge vision systems, or prototyping an interactive robot, I follow the same approach: understand the problem, work within constraints, and build solutions that create tangible impact.

02 PROJECT OVERVIEW

Edge AI Vision Deployment for Detect-and-Avoid (DAA) Systems

Lightweight object detection deployment on resource-constrained onboard platforms.

OVERVIEW

Deploy deep learning object detection models onto embedded onboard computers for real-time aircraft detection while balancing accuracy, latency, and hardware constraints.

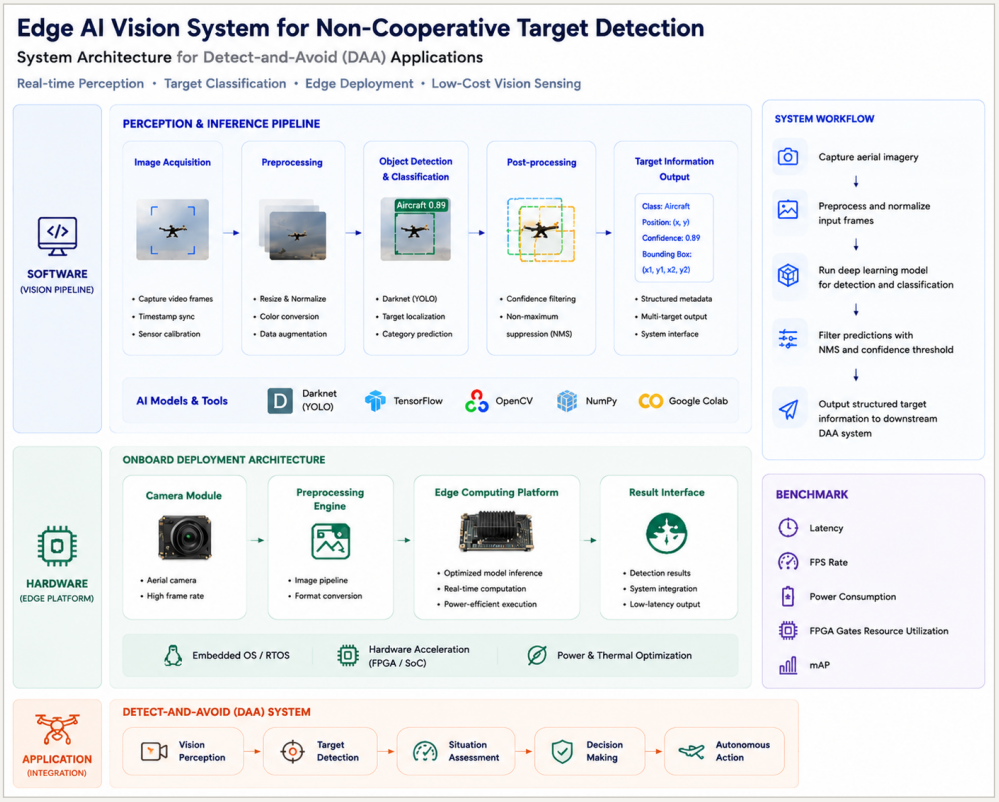
MY ROLE

Undergraduate Research Assistant

- Developed deployment pipeline from training to onboard inference
- Performed model quantization and benchmarking
- Collaborated with CV researchers to integrate newly trained models
- Standardized deployment workflow for future iterations

SYSTEM ARCHITECTURE

FIG. 01 — PERCEPTION PIPELINE / ONBOARD DEPLOYMENT / DAA



TECH STACK

Darknet (YOLO) | TensorFlow | OpenCV | Linux | Google Colab | Python

03 ENGINEERING CASE STUDY

Engineering Case Study

Deployment decisions, controlled experiments, and measured outcomes.

ENGINEERING CHALLENGES

CHALLENGE 01

Stable Embedded Deployment

PROBLEM

Deploying deep learning models onto resource-constrained onboard computers while maintaining real-time inference.

DECISION

Built a standardized deployment pipeline covering model training, quantization, deployment, and benchmarking.

TRADEOFF

Higher initial engineering effort

Significantly improved reproducibility.

CHALLENGE 02

RGB vs Grayscale Performance

PROBLEM

Grayscale images reduced computational cost but unexpectedly caused a significant accuracy drop.

INVESTIGATION

Designed paired RGB / grayscale experiments to isolate the effect of image modality. Root cause remains under investigation.

RESULT

RGB	65–70% mAP
Grayscale	30–40% mAP

DEPLOYMENT RESULTS

70.2%	14–15	Reusable Pipeline	0.82	0.76
BEST mAP	FPS ONBOARD	REPRODUCIBILITY	PRECISION	RECALL

LESSONS LEARNED

01

Embedded AI requires balancing accuracy, latency, and hardware constraints simultaneously. When applying the intelligent system, the trade-offs between different elements depend on the exact use case.

02

Standardized deployment pipelines greatly reduce repetitive integration effort.

03

Deployment is an engineering problem, not simply a model optimization problem. It requires system-level optimization across preprocessing, inference, post-processing, and hardware resources, together with continuous validation of accuracy and latency.

04 PROJECT OVERVIEW

Intelligent Robotic Pet

An ESP32-powered AI companion integrating embedded hardware, cloud intelligence, and interactive behaviors within a palm-sized system.

OVERVIEW

Modern desk workers often experience stress and isolation in long hours of focused work. This project explores how a low-cost intelligent robotic companion can provide natural voice interaction and emotional engagement through embedded AI and cloud services.

MY ROLE

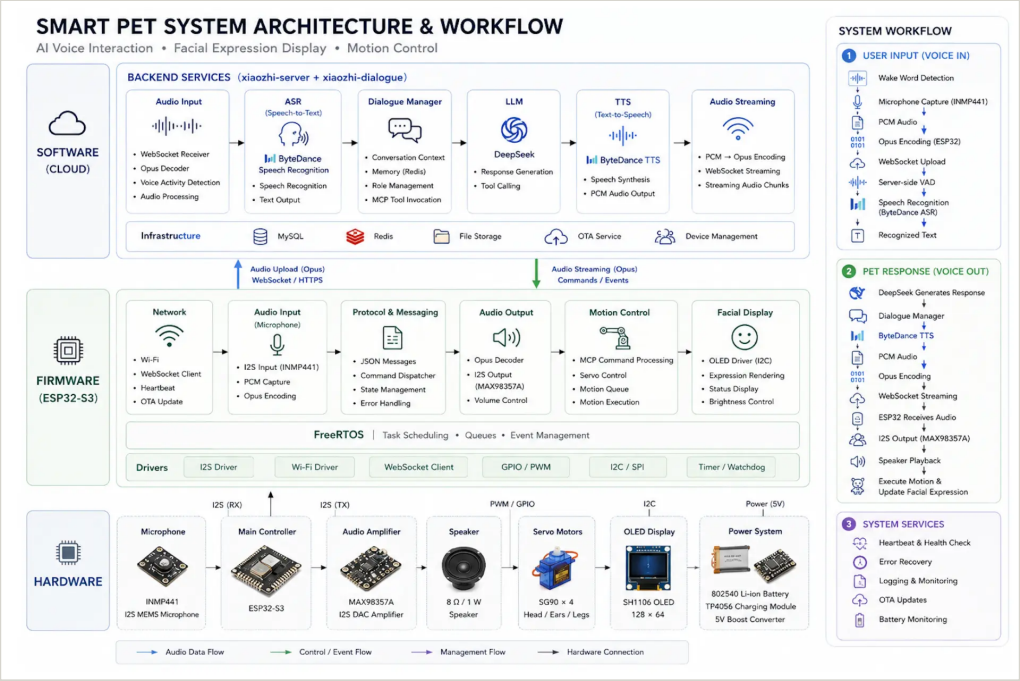
Robotics Systems Engineer

Responsible for the complete system development from prototype to deployment.

- Designed embedded hardware architecture
- Integrated Java backend with ESP32 firmware
- Developed speech interaction pipeline
- Implemented embedded audio playback
- Responsible for hardware prototyping and PCB planning

SYSTEM ARCHITECTURE

FIG. 02 — CLOUD SERVICES / ESP32 FIRMWARE / HARDWARE



TECH STACK

ESP32-S3 | Java | WebSocket | DeepSeek | Volcengine TTS | Servo Motors

05 ENGINEERING CASE STUDY

Engineering Case Study

Architecture decisions, latency optimization, and current system status.

ENGINEERING CHALLENGES

CHALLENGE 01

AI on Constrained Hardware

PROBLEM

ESP32 provides limited memory and computing capability, making it unsuitable for running modern LLMs locally while still supporting voice interaction and motion control.

DECISION

Designed a cloud-assisted architecture where the ESP32 focuses on hardware interaction while computation-intensive AI services are executed remotely.

TRADEOFF

Requires stable network connectivity

Greatly reduces hardware cost and power consumption

CHALLENGE 02

End-to-End Voice Latency

PROBLEM

The original interaction pipeline required waiting for the complete speech synthesis before playback, resulting in long response latency.

DECISION

Adopted streaming audio transmission so the ESP32 could receive and play PCM audio simultaneously.

RESULT

Reduced perceived response latency and created a more natural conversational experience.

SYSTEM STATUS

- ✓ Wi-Fi communication
- ✓ Java backend
- ✓ WebSocket streaming
- ✓ Streaming PCM playback
- ✓ ESP32 firmware integration
- ✓ Hardware prototype completed

LESSONS LEARNED

Building intelligent robots requires hardware and software to be designed together rather than independently. Many engineering challenges emerge at the interfaces between embedded systems, networking, AI services, and physical interaction.

06 PROTOTYPE & ROADMAP

Current Pet Outlook

Working hardware prototype and the roadmap toward a productized system.

CURRENT PROTOTYPE

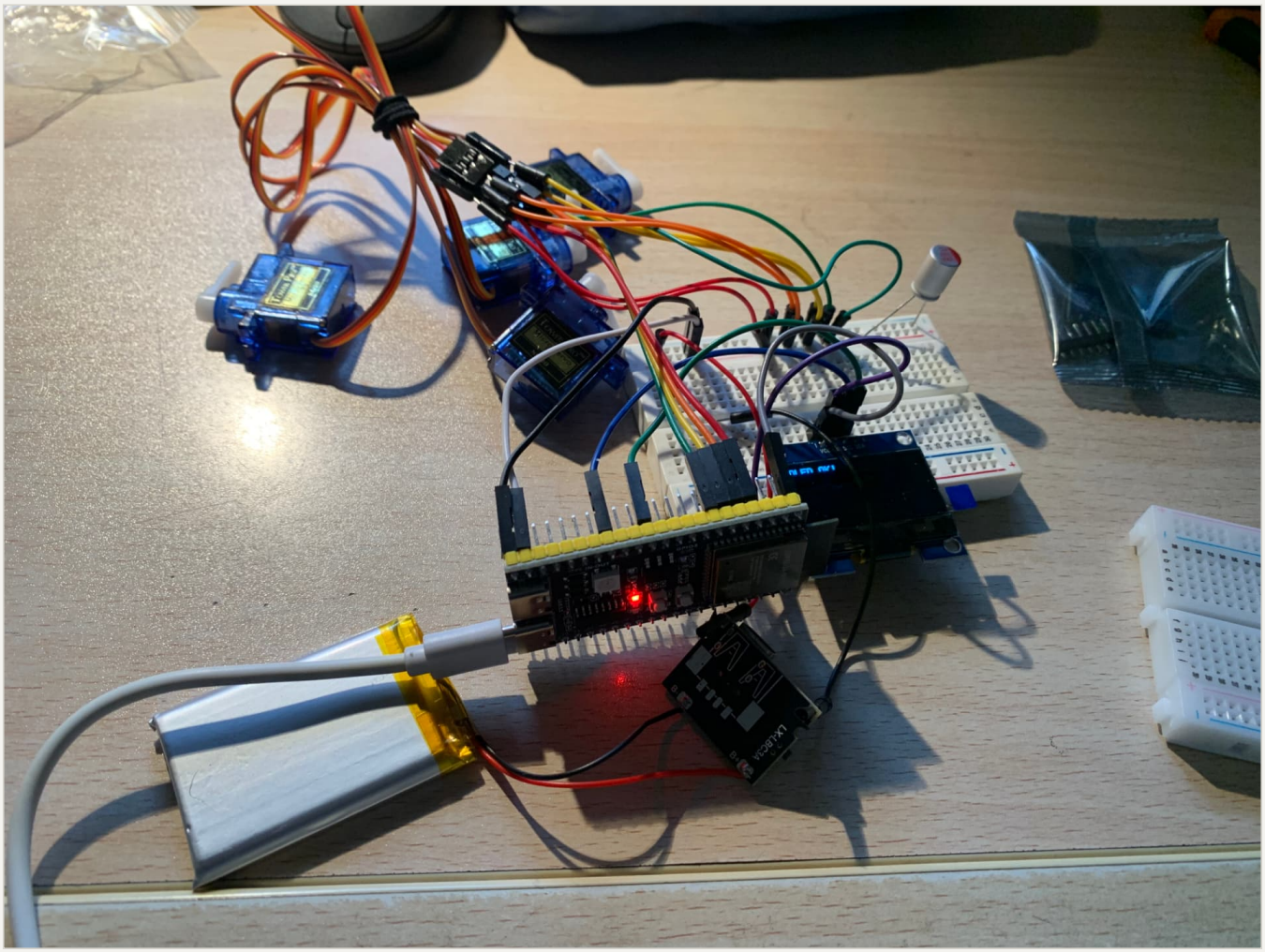


FIG. 02 BENCH PROTOTYPE / ESP32-S3 · OLED · SERVO ARRAY · LIPO POWER

FUTURE DIRECTION

01

Motion control and facial expression integration

02

PCB design and hardware miniaturization

03

Modular AI backend supporting multiple LLM / TTS providers

07 PROJECT OVERVIEW

UW-Social

A Student-Centered Event Discovery & Community Platform

OVERVIEW

Campus event information is fragmented across emails, Discord servers, social media, and organization websites. Students often spend significant time searching for opportunities while still experiencing FOMO (Fear of Missing Out).

UW-Social centralizes event discovery, community interaction, and personalized recommendations into a single platform, helping students discover meaningful opportunities more efficiently.

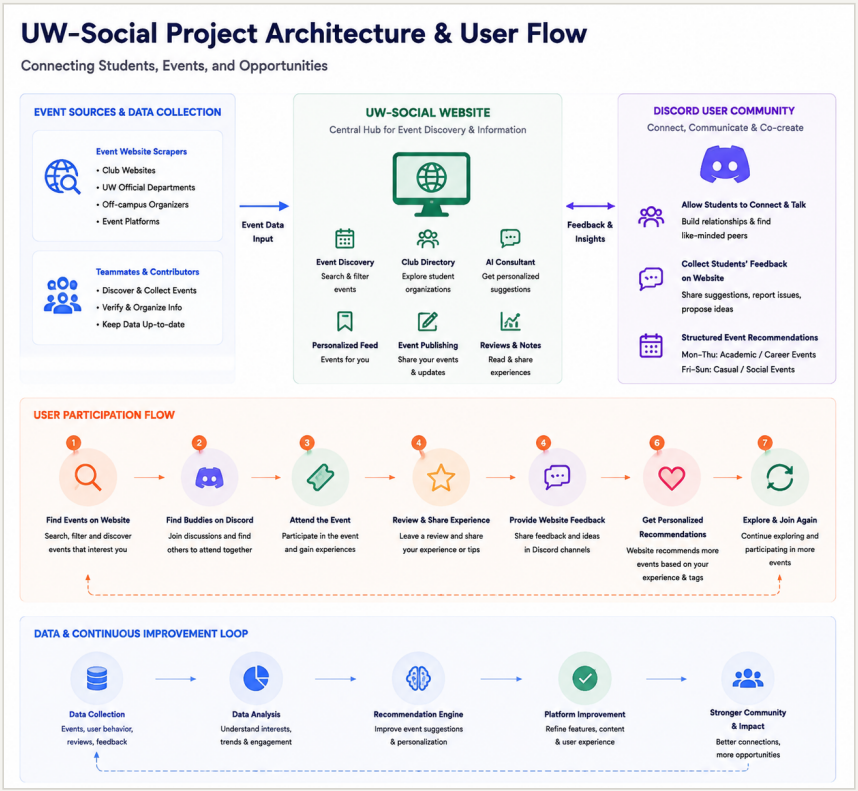
MY ROLE

Founder & Project Lead

- Founded and led a 9-member cross-functional team. Coordinated engineering, design, and marketing efforts
- Defined product vision and development roadmap. Conducted user interviews and product validation. Led iterative product improvements based on user feedback

PROJECT ARCHITECTURE & USER FLOW

FIG. 03 — PLATFORM / COMMUNITY LOOP



TECH STACK

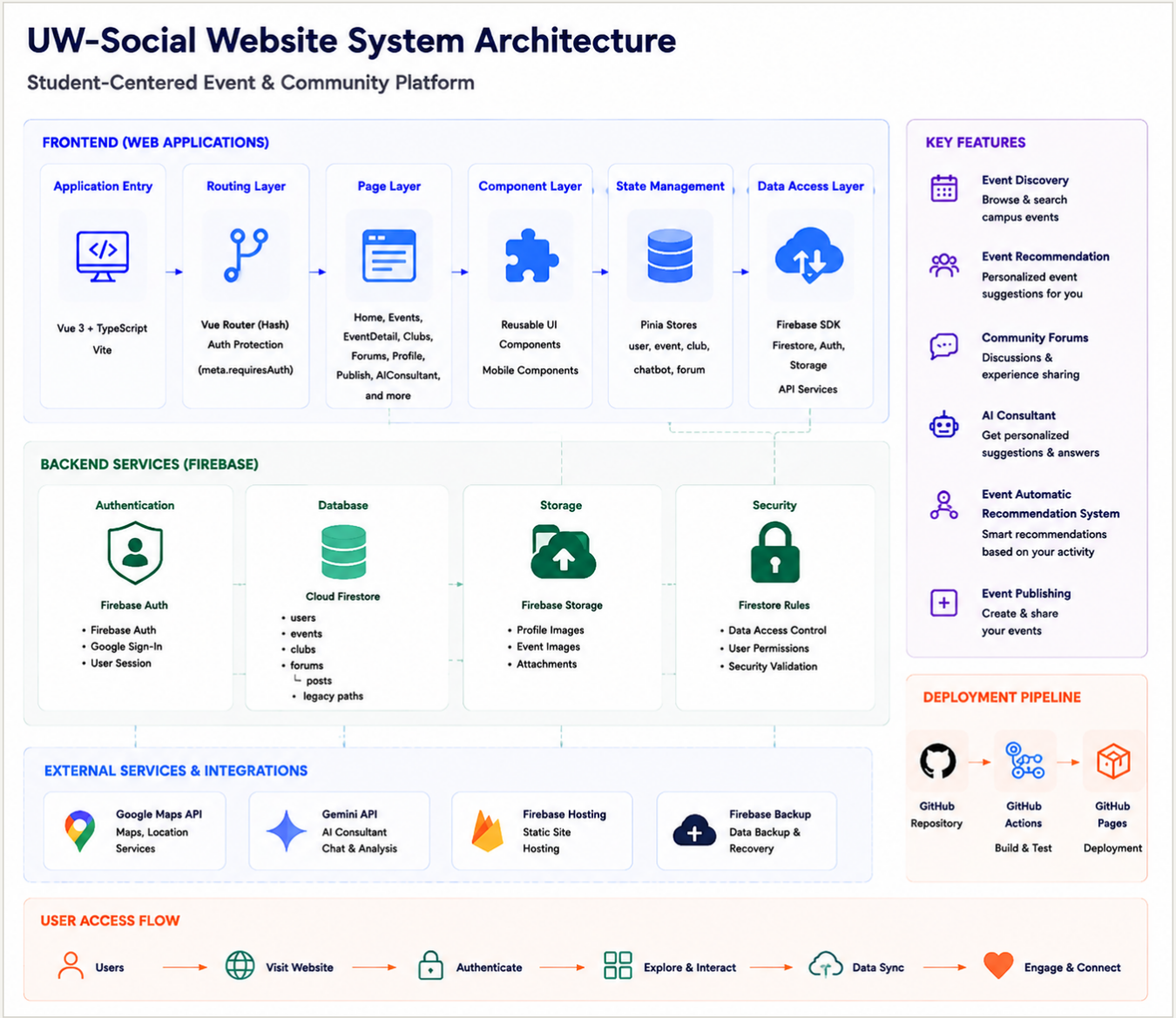
Firebase | Firestore | Google Cloud | Gemini

08 ENGINEERING & PRODUCT DECISIONS

Engineering & Product Decisions

System architecture of the UW-Social web platform.

WEBSITE ARCHITECTURE FIG. 04 — FRONTEND / FIREBASE / INTEGRATIONS



09 ENGINEERING & PRODUCT DECISIONS

Engineering & Product Decisions

Cold-start strategy, event collection, and growth outcomes.

ENGINEERING CHALLENGES

CHALLENGE 01

Solving the Cold-Start Problem

PROBLEM

Without enough event sources, students would not find value in the platform, making it difficult to attract early users.

DECISION

Focused on engineering and business events instead of attempting full campus coverage. Built relationships with student organizations while collecting user-generated event content.

TRADEOFF

Lower initial coverage

Higher information quality

CHALLENGE 02

Scalable Event Collection

PROBLEM

Most event websites limit large-scale web scraping, making automatic event aggregation unreliable.

DECISION

Collected newsletters through subscribed email accounts and converted them into structured event data.

TRADEOFF

Additional preprocessing

More stable long-term event collection

GROWTH METRICS

300+	100+	20+	60+	150+
CURATED EVENTS	ACTIVE USERS	EVENT NOTES	DISCORD MEMBERS	WECHAT COMMUNITY

LESSONS LEARNED

Successful products are built through continuous user feedback rather than feature accumulation. Solving the right problem is often more valuable than building more functionality.