Group number:
Team Members: Andrew Permatigari (862366701), George Babik(86233984), Micheal Abutin(862400100), Micheal Wong (862317955)

Use Case

Industry: Entertainment / Gambling Technology

Casinos are constantly seeking innovative ways to enhance player engagement while maintaining profitability. Traditional roulette tables offer fixed payout odds that do not consider player behavior, engagement levels, or betting patterns. Our project addresses this limitation by proposing a smart roulette table that leverages edge computing to analyze player input and dynamically adapt the payout odds. This allows the casino to encourage players to continue betting by offering temporarily favorable odds early in a session while progressively shifting the advantage back to the house over time.

Challenges Foreseen:

- Ensuring system responsiveness under 200 ms to preserve game flow
- Accurate detection of various betting patterns in noisy, real-world environments
- Maintaining transparency and fairness despite dynamic odds adjustment
- Synchronization between physical components (wheel, sensors) and computational nodes
- Preventing players from manipulating the game through the sensors or odds adjustment
- Managing downtime with the roulette table

Solution

We propose an AI-enhanced, edge-computing-powered roulette table that processes player behavior locally through embedded sensors and microcontrollers. The system will:

- Collect and process betting inputs using load cells and optical sensors
- Adapt payout odds based on detected engagement trends (e.g., increasing bets, frequent wins/losses)

- Provide real-time feedback to both players (via LEDs) and operators (via dashboards)

The broader vision goes beyond this proof-of-concept:

- Integration with centralized casino systems for analytics and historical player data
- Cloud-backed decision refinement, using aggregated data to continuously improve odds-adjustment algorithms
- Scalable deployment to other casino games using similar architectures

Demo

For our Phase 1 proof of concept, we will build a scaled-down, functional roulette system that includes:

- A player interface panel with interactive betting zones
- Real-time input detection using Raspberry Pi or Jetson Nano
- Dynamic LED feedback to display changing odds
- A spinning mechanism with real-time win/loss output
- Simulated edge-to-edge communication with a central server node that logs behavior

The demo will show that the system:

- Detects bets instantly
- Adjusts odds based on session activity
- Maintains performance under expected load

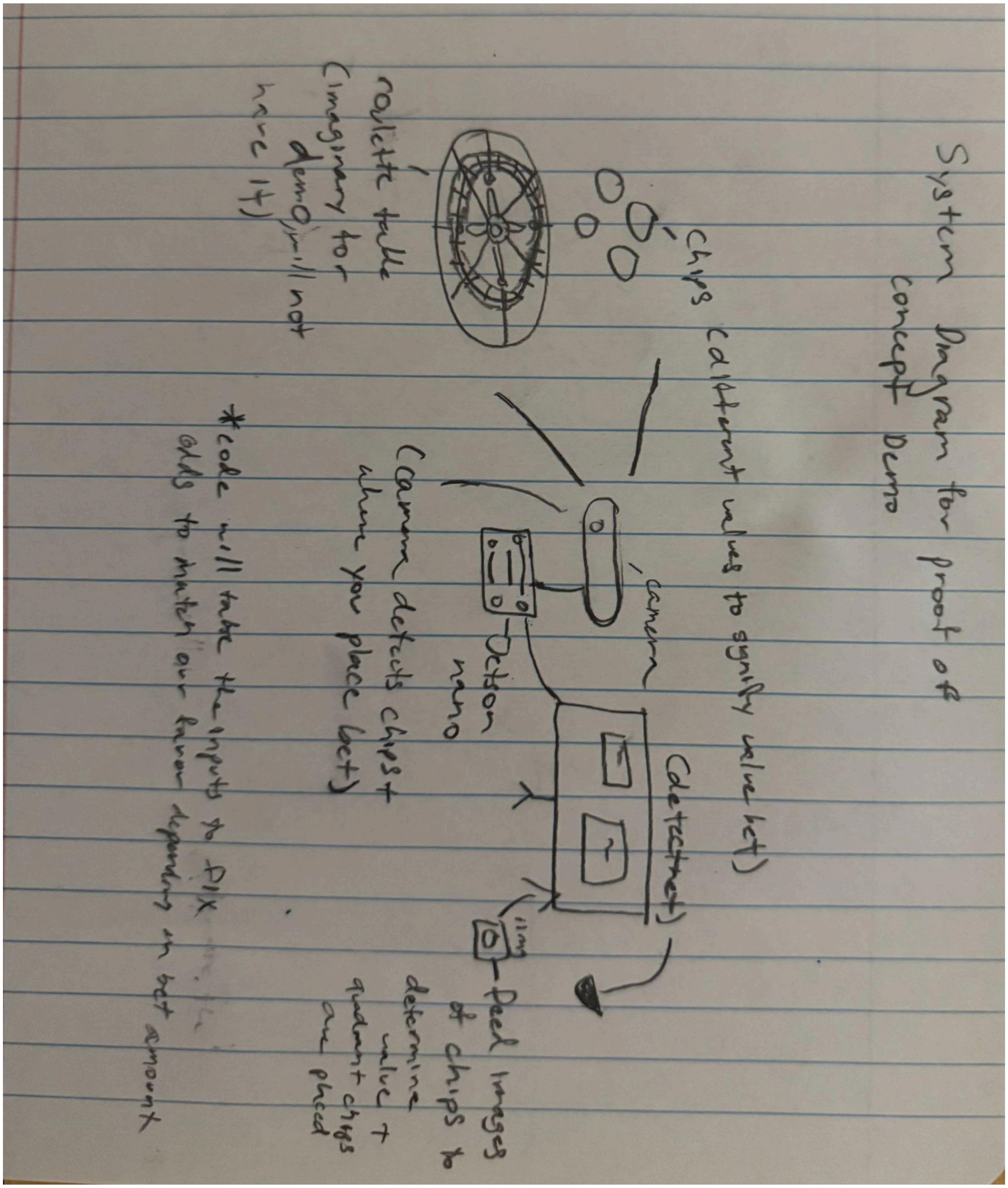
Task Distribution

Our system will follow a 3-layer architecture: Edge Device, Fog Layer, and Cloud (Optional).

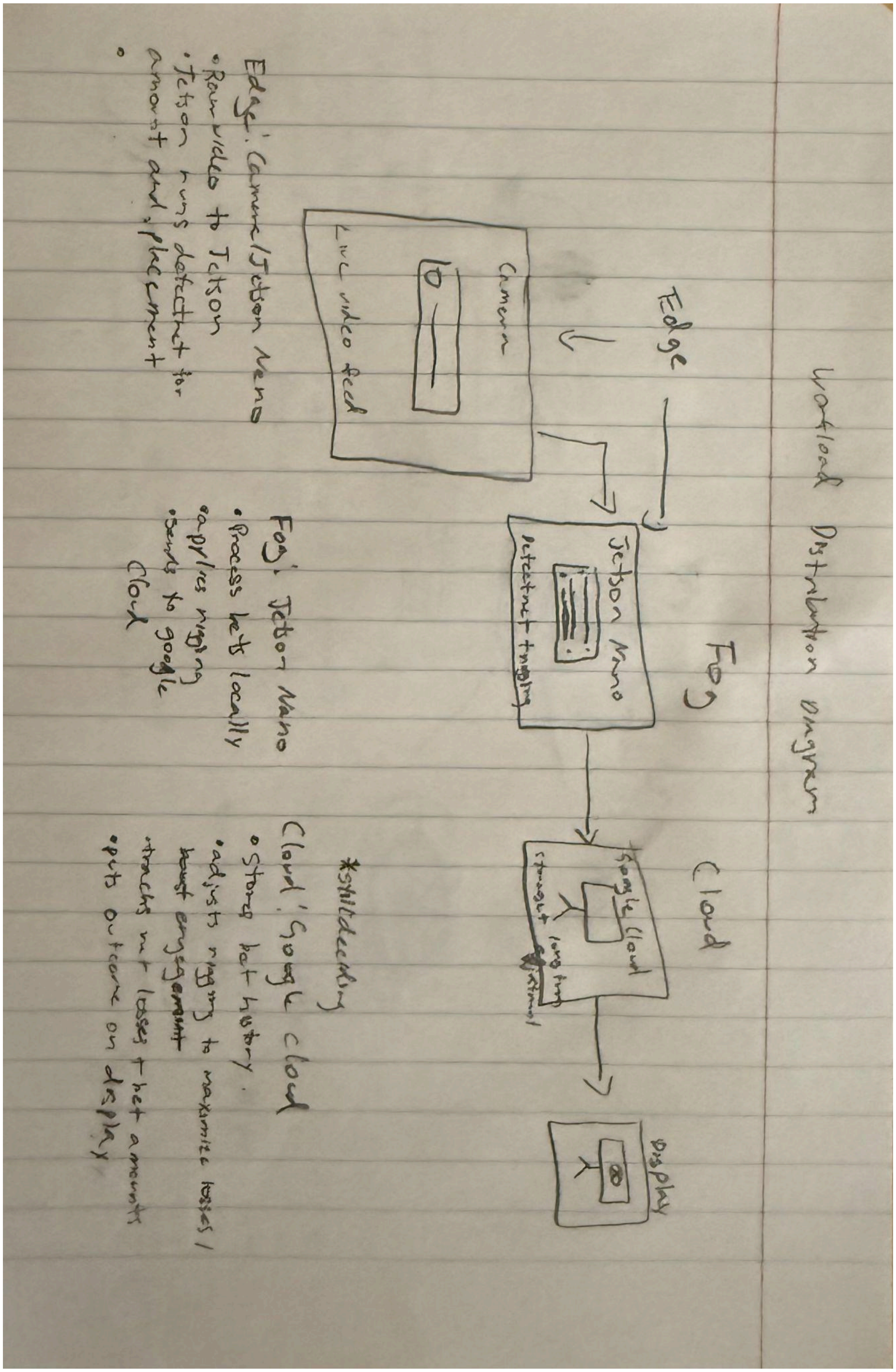
Layer	Component	Responsibilities

Edge	Raspberry Pi / Jetson Nano	- Detect player input (load cells, optical sensors) - Adjust payout odds in real-time - Control LEDs for immediate feedback
Fog	Local Server / Laptop	- Aggregate data from multiple tables - Visualize game and profit data - Buffer data to cloud
Cloud (Optional)	AWS / Firebase	- Perform trend analysis across sessions - Refine odds algorithms - Long-term player behavior analysis

System Design Diagram (for Proof of Concept Demo)



Workload Distribution



Technical Worksheet (Tool Inventory)

- Jetson Nano (1 or 2)
- Web Camera
- Google Cloud database
- Monitor
- Poker Chips

Scrum Report

Task Title	Task Owner	Estimate Amount of Work in Hours	Completed hours	Remaining Hours	Sprint	Start	Due		% of Task Completed
System Design Diagram	Andrew P, Michael Wong	3	3	0	1	5/06/25	5/09/25		100%
System Workload Diagram	Andrew P, Micheaal Abutin	2	2	0	1	5/07/25	5/09/25		100%
Github	Andrew P, George Babik	40	1	39	1	5/09/25	6/06/25		0.025%
Technical Worksheet	Michael Wong, George Babik	1	1	1	1	5/09/25	5/09/25		100%

Repository

https://github.com/apermatigari/CS131FinalProject