Goals:
The main goal of this project is to be able to provide live monitoring and reporting of cash activities generated by arcade games,

Budget:
See reward information below.

Scope:
The entire project will consist of smaller, modular projects that would integrate to offer a complete reporting solution. milestones as follow:

1) Record:
   a) Use hardware to count/ translate electric pulses generated by two input sources into usable data that could be manipulated and exported.
   b) tag all monitored activities with an RTC timestamp (see hardware considerations - rtc), and player ID information (see hardware considerations - rfid)

2) Export:
   a.1) setup integration with ioT reporting cloud service (ie. firebase.?)
   a.2) export all tagged activity to a ledger / log
   b) cron job to scan for changes to “ledger” every X seconds
   c) publish changes to database

3) Display:
   a) ”tagged” activity should be accessible via a web interface that includes the event information, along with an overall picture of the game performance.

Estimated time of completion:
30-days from the contractual date.

Hardware:
This project requires a debian-based OS capable microcontroller with wifi. As such, the raspberry pi zero w is the smallest, most cost efficient hardware solution available that fulfills both requirements. I have obtained the following components:

Raspberry pi w zero OR SIMILAR (main microcomputer board) -
2X Arduino nano - dev board to perform pulse counting (IN - OUT)
RTC Breakout - DS3234 - Real time clock IC for accurate time keeping support
HilLetGo RFID card sensor/module - player ID tagging.

Security considerations:
If at all possible, disk / partial encryption should be implemented to protect all custom running code.

Algorithm logic walkthrough:

Considerations:
*A “Device ID” is assigned to every game, and represents a fully completed hardware box solution (see hardware considerations)

*GPO Pin11 (or any) on Arduino device is assigned as MONEY_IN. PIN11 is hooked in line to receive the pulses sent from the Bill acceptor to the game. These pulses are originally designed to tell the game how much money to add as playable credit

*GPO PIN13 (or any) on Arduino device is assigned as MONEY_OUT. PIN13 is hooked in line to receive the pulses sent from the game to the printer. These pulses are originally designed to tell the printer the player is cashing out ,and a payout ticket is printed.

a) Player scans their RFID card to initiate gameplay. An RFID variable “user_rfid=11223344” is created. If no RFID card is read or recognized all subsequent tagged activity will use a “no player id present”

b) Player inserts $1 into bill acceptor

c) Bill acceptor generates an electrical pulse and sends it to the game

d) Arduino pin11 intercepts the pulses, and assigns a cash value to the pulses; i.e. 100 pulses = $1

e) An event is created and logged with a time stamp : i.e.
Device_id = 001, money_in =$1, money_out =$0, user_rfid=11223344, time=00:00:00, date=MMDDYYYY
Inversely, if/when the player cashes out, PIN13 intercepts pulses, assigns a cash value, and logs the event:
Device_id=001, money_in=$0, money_out=$1, user_rfid=11223344, time=00:00:00, date=MMDDYYYY (at the moment a cash out even is detected, user_rfid is nulled)

f) All these events are recorded on a log/ledger

g) A Cron job like process scans this log/ledger for changes X seconds, and publishes changes to database

h) Database contains all logged events, plus an overall view of the Device_ID health (using database data manipulation) i.e. (Total Money_IN 30 days), (Total Money_out 30 days),, (Total money_in daily), (Total money_out daily) each Device_ID is to have its own reporting.

g) admin logs into a web interface and sees live reporting (time stamped events), and overall Device_ID health (totals for the day/ month)

Future implementations: (BONUS)
Outside the scope of this project but future implementations include

- Lte communication for publishing outside of local networks

Hardware information:

As briefly mentioned, we will be capturing pulse activity and converting to real data. In the link below, you will see how the printer we use is using the data fed from the game to interpret these pulses.

https://na.suzohapp.com/images/pdf/70-1336-00_manual.pdf