

# NKID Arcade Management Tool

Software Specification

August 2016

## Machine properties

Name	Values	Notes
ID	i.e. NK3256PXC	A Unique Alphanumeric ID for each machine or object (walking animal)
QR code	IMAGE	A QR image encoding the ID. (the QR code is auto-generated by the system)
Name	I.e. Safari Ranger	The name of the machine or object (i.e. Walking animal #3)
Status	ACTIVE or INACTIVE	Active: machine is in use Inactive: machine is not in use and requires attention from technician
Pictures		Can have 1 or more pictures, i.e. picture of machine, PCB, cables, spare parts, etc.)
Files		Files or documents related to this machine, i.e. Ghost file, pdf files, images file or manuals.
ticket_payouts	YES or NO	If machine give out tickets when game over
Machine_categories	<ul style="list-style-type: none"><li>• Walking animals</li><li>• Claw games (prize machine)</li><li>• Redemption</li><li>• Rides (bumper cars, boats...)</li><li>• Racing</li><li>• Air Hockey</li><li>• Shooting</li><li>• Sports</li><li>• Classic Mix</li><li>• Dancing</li><li>• Music</li><li>• Fishing</li><li>• Foosball</li></ul>	Staff can select from list, a machine can have multiple categories

Location		The center ID where the machine is currently deployed
Notes/Comments		Any notes or comments if have
Dimension (HxWxD)		Dimension of the machine (Height x Width x Deep)
Vendor/Supplier		The vendor information of the machine
Play instruction		Instructions on how to play the game
Max Ticket payout		Number of tickets the machine can give out per play, i.e. 500 tiniDiem
Cost to play		How many coins is required to play, i.e. 3 Xu
Common issues		Problem or issues often occurs with this machine or type of machine, i.e. token stuck

## Design concept

### Grouping

Due to the similarity or even equal machines it is better to put them into group, this way we can save a lot of work when creating a new machine, for example we can define a machine group “Walking Animal” with common properties such as

- Name
- Ticket payouts
- Dimension
- Vendor
- Play instruction
- Max Ticket payout
- Cost to play
- Common issues
- Etc.

When creating a new machine we can assign this machine to a machine group, then all common properties of this group will be copied over, however, for some or all of the fields we

can overwrite if necessary, for example we have 2 machines from same group, Walking Animal #1 and Walking Animal #2 but the machine can have different pricing to play so we can go ahead and overwrite the group “Cost to play” of the machine we want to change.

## **Versioning concept**

For some properties or set of properties we need to keep track of changes, i.e. if the machine is faulty and we’ve manage to repair it successfully then this issue + solution needs to be documented and persisted so we can have a history of each problem and solution.

Following attributes/properties need to be logged:

- Causes (description of problem/ issue)
- Solution (description of how problem was solved)
- Datetime (auto-generated date & time when it was fixed)
- User (current user thats logged in)
- Location (current center the machine is deployed, this allow us to see the history where the machine has travelling)

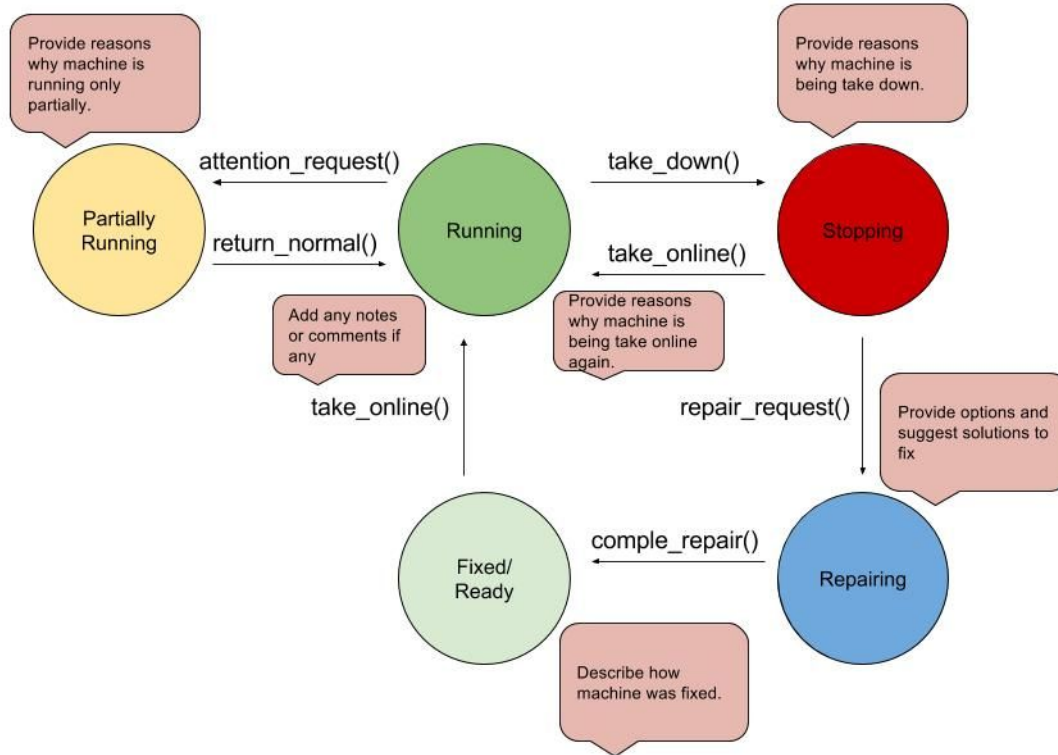
## **State machines - State management**

*(will discuss with Thach later)*

We can use simple state machine manage the stage of machines, i.e. there can be as little as 3 states:

- Running (Active)
- Stopping (Inactive)
- Waiting for Repair
- Fixed

The user can change the state of the machine from Running to Stopping when it’s been set to INACTIVE, then if there is a plan to fix the machine then user can set to “Waiting for Repair”, when the problem is fixed, the user can then set to Running state, however at this moment it is not set to be ACTIVE yet, the user or manager has to explicitly change from INACTIVE to ACTIVE.



## State flow

From state	To state	Example - case	Attributes to be saved
Running	Stopping	If machine was running but now has some issues, the technicians can decide to take it down.	<ul style="list-style-type: none"> <li>State_id</li> <li>machine_id</li> <li>current_date</li> <li>From_state</li> <li>To_state</li> <li>Description</li> <li>Author (cur user)</li> </ul>
Stopping	Running	If machine was stopped due to some operational reasons or decision by ops team, i.e. temporarily stop to do cleaning, then machine can be take online again without having to go through repair.	<ul style="list-style-type: none"> <li>State_id</li> <li>machine_id</li> <li>current_date</li> <li>From_state</li> <li>To_state</li> <li>Description</li> <li>Author (cur user)</li> </ul>
Stopping	Repairing	When machine is stopped and a	<ul style="list-style-type: none"> <li>State_id</li> </ul>

		technician is assigned to repair the machine.	<ul style="list-style-type: none"> <li>• machine_id</li> <li>• current_date</li> <li>• From_state</li> <li>• To_state</li> <li>• Description</li> <li>• Author (cur user)</li> </ul>
Repairing	Ready/Fixed	When machine is completed reparation and now ready for taking online again.	<ul style="list-style-type: none"> <li>• State_id</li> <li>• machine_id</li> <li>• current_date</li> <li>• From_state</li> <li>• To_state</li> <li>• Description</li> <li>• Author (cur user)</li> </ul>
Ready/Fixed	Running	Machine has been taken online.	<ul style="list-style-type: none"> <li>• State_id</li> <li>• machine_id</li> <li>• current_date</li> <li>• From_state</li> <li>• To_state</li> <li>• Description</li> <li>• Author (cur user)</li> </ul>
Running	Partially Running	Machine is still working but has some minor issue, i.e Machine has 4 players but only 2 players can play and 2 needs to be replaced or repaired, i.e. Ticket dispenser replacement or Coins acceptor replacement, etc.	<ul style="list-style-type: none"> <li>• State_id</li> <li>• machine_id</li> <li>• current_date</li> <li>• From_state</li> <li>• To_state</li> <li>• Description</li> <li>• Author (cur user)</li> </ul>
Partially Running	Running	When finish replacement or repair and machine can be taken fully online again.	<ul style="list-style-type: none"> <li>• State_id</li> <li>• machine_id</li> <li>• current_date</li> <li>• From_state</li> <li>• To_state</li> <li>• Description</li> <li>• Author (cur user)</li> </ul>

## Sample json node

### machine\_state:

```
{
  "state_id": "sadasd343234",
  "machine_id": "MC23455",
  "current_date": "10/06/2016 10:30",
  "From_state": "RUNNING",
  "To_state": "STOPPING",
  "Description": "Sensor ABC is faulty and need to be be fixed ...",
  "Author": "tan.duong@nkidcorp.com"
}
```

## Exporting features

The export functions allow the user to export a game list as PDF or Excel for a particular center or all centers.

## User VS. Roles

Activities	Admin or owner	Regional Technical Leader	Technician	Description
Add center/location	✓	✓		
Edit center	✓	✓		
Delete center	✓			
Add Machine group	✓	✓		
Edit Machine group	✓	✓		
Delete Machine group	✓			
Add machine	✓	✓	✓	

Edit machine	✓	✓	✓	
Delete machine	✓			

## QR code usage & concept

The reason for the QR code and unique ID to identify each individual game machine is to enable quick navigation to that particular machine via a URL, for example the QR code could be used to link to a URL, i.e.: <http://arcade.nkidcorp.com/ab34xz> , similar to a Youtube link.

## Database tables or JSON object

### Location/center

- ID
- Name
- Address details
- Center Manager info

### Location Group

- group\_ID
- Name (i.e. Zone 1)
- Locations (array of location ID)

### Vendor/Supplier

- ID
- Name
- Contact details

### Machine Group

- Group\_ID (1234)
- Name (Juno Hoops)
- Ticket payouts (true)
- Dimension (3x5x8)
- Vendor (ref. To Vendor ID)
- Play instruction (put coins and push buttons ...)
- Max Ticket payout (500)
- Cost to play (3 txu)
- Common issues (Tickets get stuck)

## Machine

- Machine\_ID (PD3218)
- Name (Juno Hoops #3)
- Ticket payouts (true)
- Dimension (3x5x8)
- Vendor (ref. To Vendor ID)
- Location (ref. To location ID, i.e. AEON Binh Tan)
- Play instruction (put coins and push buttons ...)
- Max Ticket payout (500)
- Cost to play (2 txu)
- Common issues (Tickets get stuck)

## User

- user\_ID (i.e. [phong.vo@nkidcorp.com](mailto:phong.vo@nkidcorp.com))
- password
- user\_Name (Phong Vo)
- User\_role\_ID (ref. To user role ID, ie. Regional Technical Leader)
- Locations (location\_Group\_ID)

## Recommended Technology to for the project

Following is a list of highly recommended technologies that can be used to build the Arcade Management tool.



## Platform

Google Firebase as General Technology & Libraries

- **Real-time Database:** Firebase real-time DB (JSON based NoSQL)
- **Hosting:** Firebase Hosting
- **File Storage:** Firebase storage
- **UI:** AngularJS + AngularFire (JS library for integration with AngularJS)
- **Primary Programming Language:** JavaScript

## Mobile version

...open, but considering Android App with heavy usage of the Firebase Platform & integration