

# TIKI INVENTORY PROJECT

## Description

This project consists of setting up an item inventory system based on tiki trackers by the help of the UID technologies such as QR Code, BarCode and RFID. Having a space that allows the administrator to manage in a simple way, the inventory of books in a library; for example, by attaching a QR Code or any other unique identifier to an item. The system also allows users to read information related to an article:

### Two Implementation cases :

**1. Web Application :** We have a Tiki instance installed on a server to manage the system, where the admin creates users, permissions and defines the tasks each user can do. For each item we will therefore have the possibility of generating a QR CODE containing the details of the item

**2, Mobile & Web :** We retrieve the data from different endpoints coming from the API(s) and thus, these same APIs also make it possible to communicate any other Hardware architecture, each user/hardware retrieves and processes the data according to access rights defined by the admin.

### In short, we will have to carry out the approaches below :

- Specific trackers for categorizing and monitoring stock and/or other information collected
- Encode unique identifiers of items on Barcode, QR codes or RFID chip
- Make available a visual and intuitive interface eg. QR Code scanner for status update of a given item
- Set up a procedure for automatically updating the date of the last operation on an item
- Make an interface available to view the items in a given group i.e stock

## USER STORY

- Management of inventories, permission and creation of trackers on the admin side will be available on the interface of the tiki instance
- An operator interface will be available on mobile application (Ionic Vue) or web application, all connected to the main tiki instance
- Have Tiki communicate with material objects (Internet of Things) using Tiki trackers, NB: in the case of scanning QR codes or Bar codes, the material object being the camera of the operator's smartphone, will be able to benefit directly from the same connection available on the smartphone.
- Automate various operations in the agricultural, industrial/factory, automotive etc. sector by carrying out permanent communication between the connected hardware devices and the Tiki trackers;
- Manage the deployment of IT and networking equipment required for a building (for 2023 or 2024)

# TO-DO LIST

1. Graphic Design (Create a PENPOT file) for the Progressive Web App User Interface;
2. Web App User Interface based on the VueJS SSR framework in which we will have:
  - Security: Login & Registration
  - Security: Management of authentications, users and access levels
  - Retrieval and display of items from Tiki trackers
  - Display of statistics;
  - Update information of an item;
  - Scan: decryption of data in a QR code and information recovery
  - Inventory management (Admin)
  - Sorting, Ranking, Grouping, Deleting, Adding and updating Items;
3. A Vue Progressive Web App (PWA) with the same functionality as (1) but all API-based. NB this same app will be compiled in hybrid by Ionic
4. In the Future : A library (for the Internet of Things) named **TikiThings** : A library that helps to easily establish a communication between the TikiTrackers and the microcontrollers to facilitate the integration of Tiki APIs and TikiTrackers with Hardware Devices.

# TASK ASSIGNMENT

**These are major tasks, they will be re-divided into small tasks to ensure that we do a sprint each week, so we adopt to work in a SCRUM way.**

- Prepare the environment and model the trackers
- Create a skeleton app interface
- Change the skeleton into wiki pages
- Translate the model into SFC VueJS
- Integrate frontend with API calls
- Create an admin wiki page to generate and print items Unique QR code per item
- Integrate QR code reading mechanisms
- Implementation
- Testing
- Publishing Process

# TIKI AND HARDWARE

# COMMUNICATION ( TikiThings / IoT)

\_\_1. The TikiThings library: In order to communicate with the trackers, the IoT / Electronics programmer will have to integrate our library by following the documentation and the structure of the API such as:

- An endpoint to manage the authentication: It must have created (generated) an auth-key that it will use to access data available on other endpoints
- [GET](#) One or more endpoints to generate the list of data to which the user has access;
- [POST](#) Endpoint(s) to store different data in real-time trackers
- [DELETE](#) Endpoint(s) to delete data in trackers;
- [UPDATE](#) Endpoint(s) to update data in trackers;
- **COMMANDS**: Create script commands, allowing communication between software and hardware (start, sleep, lock, check, use, connect, restrict, etc.);
- **RFID READING**: one or more endpoints to verify the authenticity of the RFID card and the data associated with the card in the Tiki trackers;
- Management of hardware authentication

## 2. Hardware Authentication Management

## 3. API Documentation

# Progressive Web App (PENPOT DESIGN)

# Native Mobile App (PENPOT DESIGN)

**See also :**

[The Figma product of the project](#)

[Project Cards on Github](#)