Cashless RFID System

**Definitions**

**TODO (Daniel)**

1. **Event:** Single / Multi day Music festivals, conferences, corporate events and so on.
2. **Attendee:** Customers. Regular people that bought, subscribed or somehow got a ticket to the event.
3. **Wristband:** An RFID-enabled wristband which represents the ticket itself and identification of the attendee. The attendee that owns the wristband has the right to get into the event. The wristband will also work as an e-wallet, the attendee will be able to top-up money into the wristband prior or during the event and use the wristband as payment method for any kind of products and services the event sells.
4. **WUID**: Wristband Unique Identifier
5. **Desktop POS:** A Point of Sale solution that runs into a PC.
6. **PDA POS:** A Point of Sale solution that runs into a handheld device.
7. **Customer:** the one buying the solution (event organizers)
8. **CRUD**: Create, Read, Update, Delete
9. **client:** the client-side in a client-to-server architecture
10. **Cashier**: Any of the authorized event's cashier responsible for handle payments
11. **Crew member**: a waiter/bartender.

**Overview**

A system to enable to cashless payments in events (e.g. Tomorrowland) via RFID technology. Each event attendee will own a wristband to which it will be possible to top-up amounts (prior or during the event). RFID readers will be available across the event so that attendees can make payments. The payment validation is via proximity only, no PIN or password is required from the attendee.

**Client-side**

There will be several client modules depending on the functionality and target platforms

1. Desktop

The desktop client is PC-based software targeting Android platform. It assumed that it will be equipped with following input/output devices:

* monitor (with touch-screen input)
* thermal print (for receipts)
* RFID reader

It's also assumed there will be a stable Internet connection available for it.

There can be cases where the internet connection is poor or unavailable, for this cases there will be an offline solution that mirrors the serverless functionality.

This client will provide the following modules:

* 1. Cashier Module
     1. User goes to the event cashier and request top-up amount;
     2. Cashier UNLOCKS the terminal;
     3. User pays the cashier;
     4. After payment is ok, the cashier requests the user to tap the wristband to the sensor;
     5. Cashier enters all the information regarding the method payment used [ Debit (Master, Visa, etc), Credit (Master, Visa, etc), money]
     6. System updates user balance.

\* there will be integration with payment systems. For example, the customer might use its own card processing device and after approval, it will manually inform to the top-up module.

* 1. Point of Sale (POS) Module:
     1. Attendee arrives to front counter and waits for service;
     2. Crew member will UNLOCK the station with his pin and ask user to tap the wristband;
     3. The system will unlock and show menu to Crew member;
     4. User will ask the products he wants and Crew member will choose at menu;
     5. Crew member asks user tap to confirm;
     6. If balance is enough, the system will update the Attendee's balance and inform the Crew member the operation was successful.
     7. Crew member handles the products to Attendee.

\* there will be no stock management

\*\* For both (A) and (B), all activities should be logged, so that clearing can be performed afterwards. The system should also log the cashier ID for auditing, along with the other transactions details.

1. PDA

This is Android-based mobile-device [which one **we need to decide based on our own needs like network capability, memory, processing capability and so on**] capable of reading RFID tags. It will provide the same POS function as the Desktop version. There won't be top-up for this client.

\* Top-Up is exclusive cashier function described in “A".

\*\* The system needs to be localized, it means, there will be a dictionary and the system can be configured to be used initially in Portuguese or English, and must be ready to run in any language that the dictionary is built.

1. Website

A web-based application for attendees to use for several purposes. The web application must be deployed into a domain controlled by the Customer. Basic customization (colors and images) must be provided in a easy way. The attendees will be to use the website to:

* 1. Sign-up/Sign-in
  2. Register their wristband
  3. Top-up (Payment Method to be defined)
     1. User access the event website and creates an account;
     2. User binds the wristband to his account using the WUID engraved into the wristband.
     3. User top-up money into the wristband via credit or debit card. ( At this point the module will need to integrate to some interface that will charge the user card and inform our system the credit is ok)
     4. User gets confirmation and the system updates the user balance.

1. Admin Website

Admin or Customer's staff will have access to web portal for different management tasks such as:

* User management (CRUD)
* wristband management (CRUD)
* Reports in general:
  + POS Sales
    - By Product
    - By POS
    - By operators
    - By different stores (Bar, restaurant, services that use cashless, etc)
  + Cashier sales:
    - Credit
      * Master
      * Visa
      * AMEX
      * ETC
    - Debit
      * By card type too
    - Cash
    - Voucher
    - etc
  + Performance
* (...) what else ?

**Backend**

1. **REST-API**

It will be the main interface with all clients. It will be RESTful web-service over HTTPs. It must provide reliable authentication and authorization options. It will interface all to the database. No client should communicate directly to the database, it will must go through some defined and supported operation of the API.

1. **Database**

A SQL-based database to persist all information required for the system.

1. **Contingence**

The contingence can be made by proxy-like server that can be a capable notebook or desktop (Linux) with stable and fast Internet connection, that can forward API requests to the remote server.

Design:

Menus and general config will load from XML files.

* POS Flow:

1. [POS\_Unlock](https://drive.google.com/open?id=1zNCqMT2NUEiAkKDsNfP5AXsUCXKSjNs-3NvlCEVCfAs)
2. [POS\_TapToStart](https://drive.google.com/open?id=17-lGr1-Yyo9txr5YxzbcTCXk9EVWX7aJ3p8QeX2JkSI)
3. [POS\_Menu](https://drive.google.com/open?id=1vTH2NPuBW_zEbMYNGEM4WUO63obi_WCOD0fIgbt5OLQ)
4. The Menu can contain sub menus as follows (Some Products can repeat and point to one register, it will be all configurable):
   1. [POS\_SubMenu](https://drive.google.com/open?id=1D2G9slRonSJktb9_zoovbbj98NeAMSzgeN3y_kzLUIk)
   2. [POS\_SubSubMenu](https://drive.google.com/open?id=1Dj7FQTdmgFe0ZU6hdKHwl42k7eSYNdBtFU9zdHYAUnY)
5. [POS\_TapToConfirm](https://drive.google.com/open?id=1nsQ_UwujaEjoGZ-axHpI1e_ATUbeuVeQZfYZzvvwxE8)

* Cashier Flow

1. [Cashier\_Unlock](https://drive.google.com/open?id=1sfkwDdYmxv9Jmzg9laQMoGvRTWFUw-oK2_pU-OdnKeM)
2. [Cashier\_TapToStart](https://drive.google.com/open?id=1fVfVP46dlzabVf6rnvQmqNahZiPL0luZuDceY3tJcJY)
3. [Cashier\_PaymentType](https://drive.google.com/open?id=19vDUx6a6NwDztmwe-kbRH2g2kD9BXzNgLzReopA3HBQ)
   1. Payment Type Cash drives direct to amount screen (4.)
   2. [Cashier\_PaymentTypeDebit](https://drive.google.com/open?id=1WkQeKRkkFq9l5VfVXDRgjl8C3SyIhKN4hHOeELqohMc)
   3. [Cashier\_PaymentTypeCredit](https://drive.google.com/open?id=14nAoJmfhMjD-_W4irAZ0s6QjbnxW9Mx5mFaoZXQrPjw)
4. [Cashier\_Amount](https://drive.google.com/open?id=1wvvkMeizksKT00t1RZSMo-eGV72ouRnECfrQlc6CsNk)
5. [Cashier\_TapToConfirm](https://drive.google.com/open?id=1aKfaTD-lcqvQxN3FzNbDy6v2SBjHrh1AcZQ2wSplrdU)

User info screen can be shown from any step after user Tap and will show user information like ballance, transaction history, products user buyed and so on. It will show up after click user picture or emot at left up corner.

[User\_Info](https://drive.google.com/open?id=1mrY2KoUDD7-b7XJXuYkVIcdFHeWgilCNkQFVa_oqjgQ)