Mô tả tính năng POS CAFÉ SUNO

1. Yêu cầu :

POS Bán hàng sử dụng riêng cho ngành hàng Café – Nhà hàng. Người dùng có thể thiết lập sơ đồ Khu vực, Bàn – Phòng trong mỗi khu vực.

Xem báo cáo doanh số hằng ngày.

Sử dụng được trên nhiều thiết bị

1. Tính năng:

* Đăng nhập
* Tạo sơ đồ phòng bàn
* Tạo đơn hàng
* Đơn hàng cho quán trà sữa và dịch vụ tính giờ
* In đơn hàng cho bếp / In tem báo bếp cho quán trà sữa
* In Bar / Bếp
* Lưu đơn hàng
* Xem báo cáo
* Đồng bộ tình trạng đơn hàng trên nhiều thiết bị.
* Kết ca

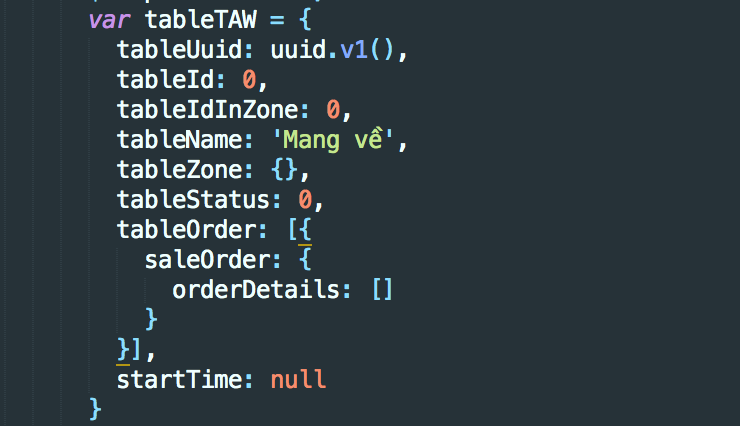
1. Tools

* Ionic version 1 : <https://ionicframework.com/docs/v1/overview/#download>
* NodeJs version 4.x
* MongoDB version 3.0.6

1. Chi tiết tính năng

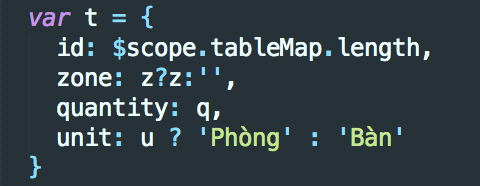
A. **Danh sách Modal**

1. Modal Table



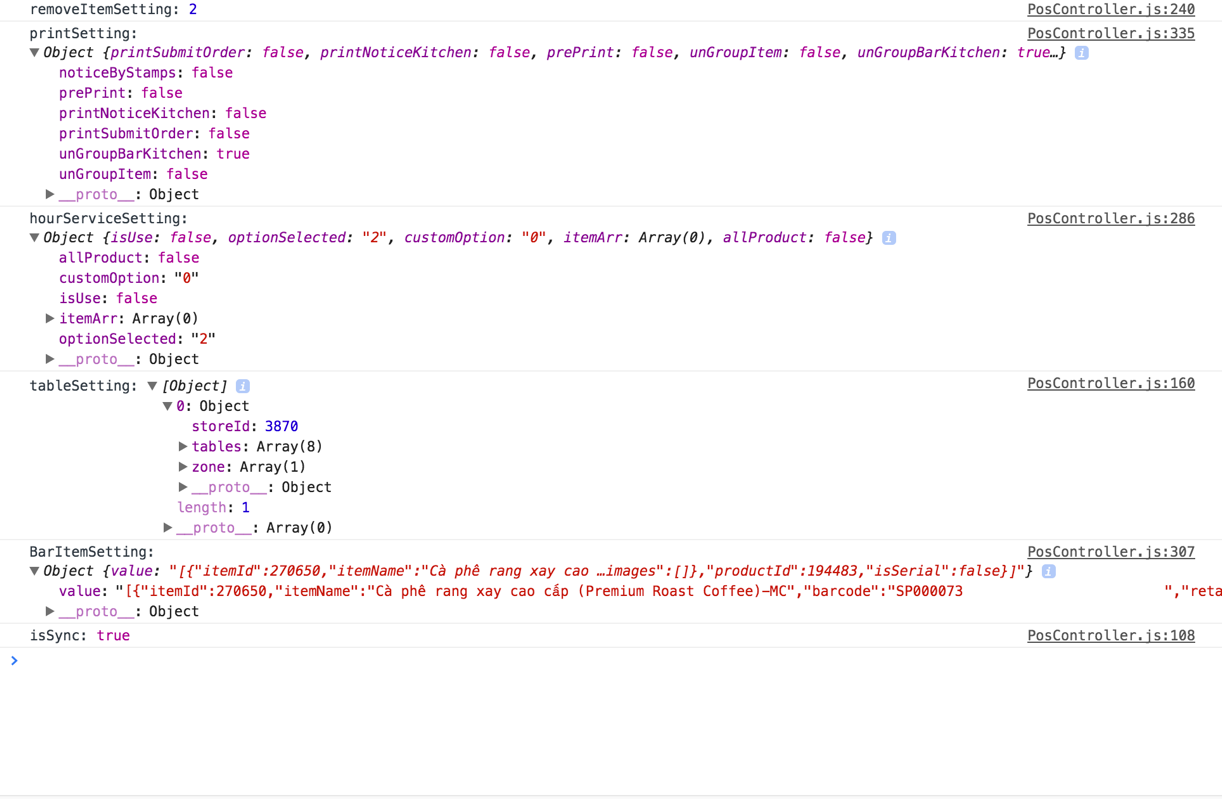
|  |  |
| --- | --- |
| **Key** | **Value** |
| tableUuid | ID unique của table, sử dụng để xác định bàn khi đồng bộ dữ liệu giữa các thiết bị. TableUuid được sinh ra khi lần đầu tiên tạo sơ đồ bàn, chỉ thay đổi khi có cập nhật sơ đồ bàn. |
| tableId | Table index |
| tableName | Được đặt theo “tên đơn vị + số thứ tự trong khu vực – tên khu vực” ( Có 2 đơn vị là Bàn / Phòng ).  Bàn đầu tiên trong sơ đồ được đặt tên “Mang về” |
| tableZone | Chứa object TableZone ( Khu vực ) |
| tableStatus | Tình trạng bàn ( 0 : bàn trống , 1 : bàn có hóa đơn ) |
| tableOrder | Các đơn hàng được tạo trong bàn |
| startTime | Ghi nhận thời gian khi đơn hàng đầu tiên trong bàn được tạo |

1. Modal TableZone



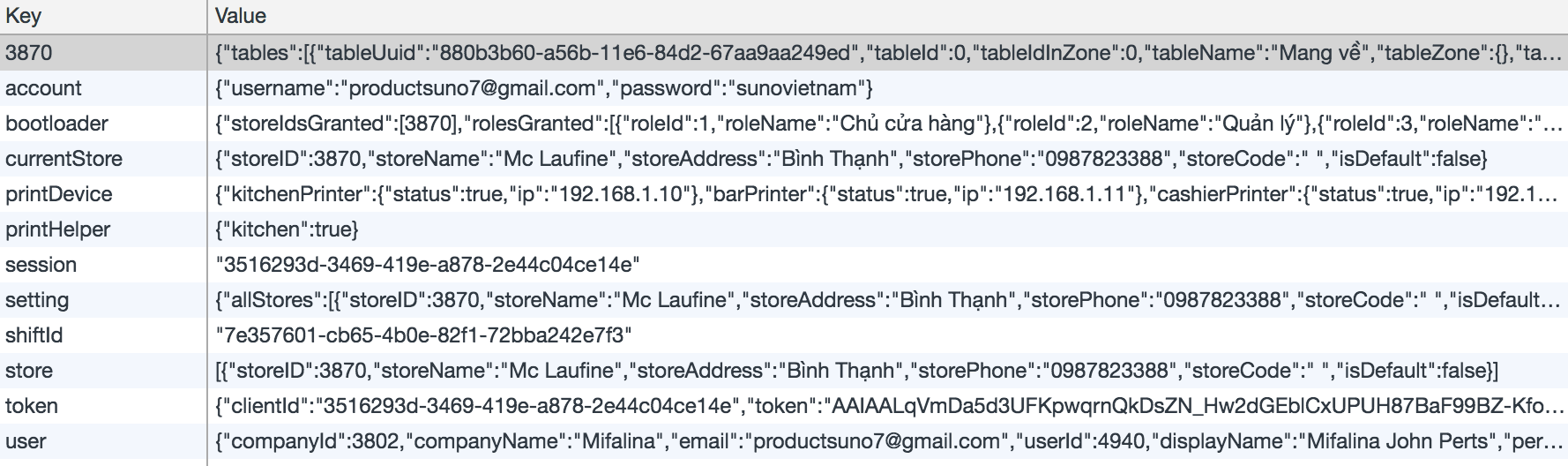
|  |  |
| --- | --- |
| **Key** | **Value** |
| Id | Zone Index |
| Zone | Tên khu vực |
| Quantity | Số lượng bàn trong khu vực |
| Unit | Đơn vị ( Phòng hoặc Bàn ) |

1. Các thiết lập:



|  |  |
| --- | --- |
| Key | Value |
| removeItemSetting | Thiết lập hủy món  1: Hủy món không cần xác nhận  2: Cho phép hủy món khi chưa in bếp / Đã in bếp thì phải xác nhận hủy món từ cấp Quản lý trở lên  3: Chỉ quản lý trở lên mới được hủy món  4: Chỉ chủ cửa hàng mới được hủy món |
| printSetting | Thiết lập in   1. noticeByStamps:In tem báo bếp ( cho trà sữa ) 2. prePrint:Thiết lập cho phép In đơn hàng tạm tính 3. printNoticeKitchen: Không In khi báo bếp 4. printSubmitOrder: Không In khi thanh toán nhanh 5. unGroupBarKitchen:Tách in Bar-Bếp 6. unGroupItem:Tách từng món trong đơn hàng (cho trà sữa) |
| hourServiceSetting | Thiết lập dịch vụ tính giờ   1. allProduct:Áp dụng cho tất cả hàng hóa 2. customOption: Số phút thiết lập làm tròn ( nếu chọn Làm tròn giá theo : Tùy chọn ) 3. isUse:Trạng thái sử dụng tính năng 4. itemArr:Danh sách hàng hóa thiết lập tính giờ 5. optionSelected:Lựa chọn thiết lập làm tròn giờ theo block 0 : Tùy chọn 1: 15 phút 2: 30 phút 3: 60 phút |
| tableSetting | Thiết lập sơ đồ bàn  mảng sơ đồ bàn được lưu theo cửa hàng  [  {   1. storeId: ID cửa hàng 2. tables:Array(Modal Table) 3. zone:Array(Modal Zone)   },  ] |
| BarItemSetting | value : [  Danh sách hàng hóa thiết lập In quầy Bar  ] |
| isSync | Trạng thái sử dụng đồng bộ |

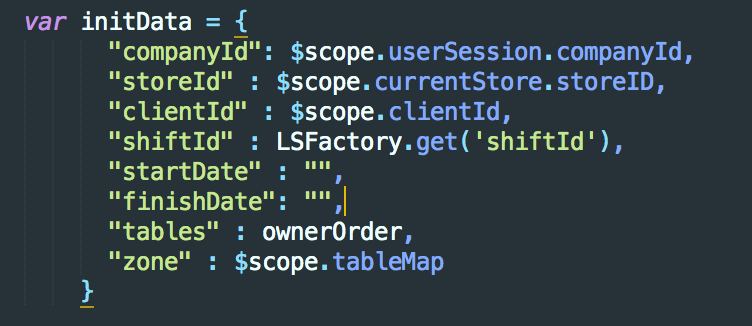
d. Dữ liệu lưu ở LocalStorage



|  |  |
| --- | --- |
| **Key** | **Value** |
| StoreID | Array( Modal Table ) |
| printDevice | Thiết lập máy in, sử dụng cho app cài trên thiết bị di động |
| printHelper | Thiết lập in nhờ, sử dụng cho desktop |
| shiftId | ID cho phiên làm việc, được sử dụng để xác định phiên làm việc của 1 cửa hàng trong quá trình đồng bộ đơn hàng |
| account | thông tin tài khoản, sử dụng cho việc ghi nhớ đăng nhập trên app |
| bootloader | Thông tin tài khoản |
| currentStore | thông tin cửa hàng đang sử dụng |
| setting | thông tin công ty |
| store | Danh sách cửa hàng |
| token | token lấy dữ liệu của user |
| user | thông tin user |

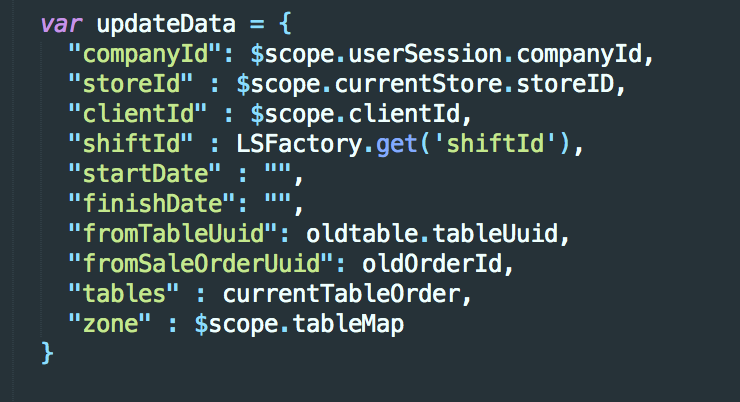
e. Cấu trúc data đẩy lên khi đồng bộ

initShift / updateOrder / completeOrder /



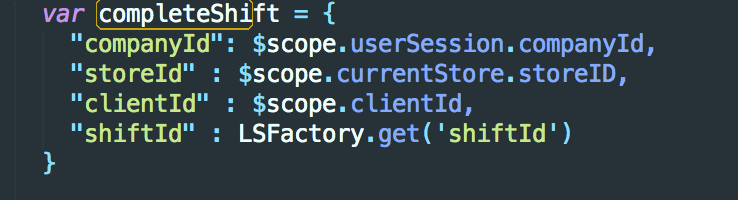
|  |  |
| --- | --- |
| **Key** | **Value** |
| companyId | Id công ty đang sử dụng |
| storeId | Id Store đang sử dụng |
| clientId | sessionId user đang đăng nhập |
| shiftId | Id phiên làm việc hiện tại của cửa hàng, lấy từ localStorage |
| tables | chỉ lấy những bàn có hóa đơn do user đang đăng nhập tạo ra |
| zone | danh sách khu vực hiện tại |

moveOrder

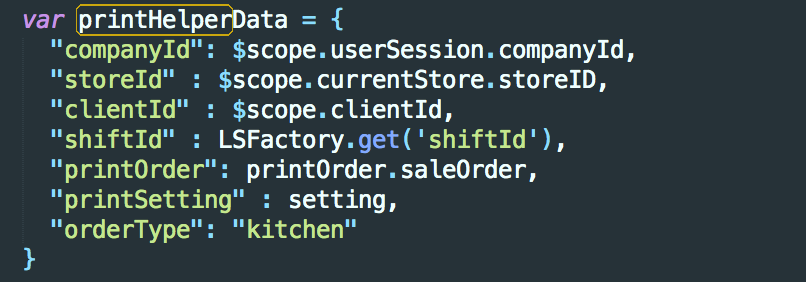


|  |  |
| --- | --- |
| Key | value |
| fromTableuuid | Bàn hiện tại đang thao tác |
| fromSaleOrderUuid | Đơn hàng hiện tại cần xử lý |

completeShift



printHelper



|  |  |
| --- | --- |
| **key** | **value** |
| printOrder | object saleOrder ( nội dung đơn hàng cần in ) |
| printSetting | thông tin company & store sử dụng trong mẫu in |
| orderType | kitchen / cashier ( cho phép in nhờ khi báo bếp & in nhờ khi thanh toán ) |

**B. Quá trình nạp dữ liệu và kiểm tra dữ liệu khi vừa truy cập**

- Load category

- Load Danh sách hàng hóa ( limit 1000 )

- Load Mẫu in

- Load Thiết lập Hủy món

- Load Thiết lập in

- Load Thông tin công ty

- Load Thiết lập dịch vụ tính giờ

- Load Thiết lập sơ đồ bàn

- Load Thiết lập in Bar / Bếp

Sau đó

- Kiểm tra thiết lập sơ đồ bàn ( theo StoreID ) được lưu dưới localStorage

->Nếu có dữ liệu, nạp dữ liệu sơ đồ bàn từ localStorage

->Nếu không có dữ liệu

->Kiểm tra thiết lập sơ đồ bàn trên server (Api)

->Nếu không có sơ đồ bàn

- Hiện popup tạo sơ đồ bàn ( nếu là quyền chủ cửa hàng )

- Tạo dữ liệu 1 bàn mang về, không tạo sơ đồ, bán hàng bình thường như Pos ( đối với quyền nhân viên ).

->Nếu đã có sơ đồ bàn

-> Kiểm tra thiết lập đồng bộ

->Nếu có sử dụng đồng bộ

Khởi tạo kết nối socket

**InitShift (\*)**

**Cập nhật thông tin đơn hàng dưới localStorage**

**-** Lấy thiết lập In hộ ( nếu truy cập bằng desktop )

- Lấy thông tin máy in ( nếu truy cập từ app )

**C. Flow đồng bộ đơn hàng**

**Điều kiện isSync = true**

**Room connect : room=companyId\_storeId**

|  |  |
| --- | --- |
| **Client** | **Server** |
| **InitShift**  (InitShift được thực hiện khi user truy cập, reload web / app).  -Gửi đơn hàng của User hiện đăng nhập đang lưu tại localStorage lên server, kèm theo ShiftId của phiên làm việc hiện tại.  -ShiftId và danh sách đơn hàng của user đều có thể trống | **InitShift**  Kiểm tra ShiftId được gửi lên từ client.  Nếu ShiftId gửi lên là null, lấy ShiftId & data hiện tại của companyId\_storeId gửi xuống client.  Nếu ShiftId ko trùng với ShiftId hiện tại trên server -> gửi xuống client exception báo lỗi invalidShift.  Nếu ShiftId trùng với ShiftId trên server, **Kiểm tra & cập nhật dữ liệu user gửi lên**, push danh sách đơn hàng mới xuống tất cả client  **broadcastOrders**  gửi xuống clients tất cả đơn hàng hiện tại trên server |
| **updateOrder**  Được thực hiện khi dưới client có một trong số các hành động  -Báo bếp  -Tách hóa đơn  -Hủy món đã báo bếp  -Ngừng tính giờ 1 item trong đơn hàng  Client sẽ gửi lên server danh sách đơn hàng của bàn hiện tại đang thao tác. | **updateOrder**  **Xử lý tương tự InitShift** |
| **moveOrder**  Được thực hiện khi dưới client có một trong số các hành động  -Đổi bàn  -Ghép hóa đơn  Client sẽ gửi lên server danh sách đơn hàng của bàn hiện tại đang thao tác. | **moveOrder**  Kiểm tra ShiftId được gửi từ Client  Nếu ShiftId trùng với ShiftId hiện tại trên server tiến hành xử lý  - Tìm và xóa đơn hàng tại bàn cũ dựa theo **fromTableUuid** và **fromSaleOrderUuid** được gửi lên từ client  - Thêm / Cập nhật đơn hàng theo dữ liệu mới đẩy lên từ client  **broadcastOrders**  gửi xuống clients tất cả đơn hàng hiện tại trên server |
| **completeOrder**  Được thực hiện khi dưới client có một trong số các hành động  -Xóa trống đơn hàng  -Thanh toán đơn hàng  Client sẽ gửi lên server danh sách đơn hàng của bàn hiện tại đang thao tác. | **completeOrder**  Kiểm tra ShiftId được gửi từ Client  Nếu ShiftId trùng với ShiftId hiện tại trên server tiến hành xử lý  - Tìm và xóa đơn hàng ra khỏi danh sách đơn hàng đang được xử lý đồng bộ trên server  - Lưu đơn hàng vào bảng History  **broadcastOrders**  gửi xuống clients tất cả đơn hàng hiện tại trên server |
| **completeShift**  Được thực hiện khi dưới client có một trong các hành động  -Kết ca cuối ngày  -Tắt tính năng đồng bộ  Sau khi gửi tín hiệu **completeShift**  lên server  - Xóa shiftId lưu dưới localStorage  - Xóa sơ đồ bàn hiện tại dưới localStorage | **completeShift**  Nếu ShiftId trùng với ShiftId hiện tại trên server tiến hành xử lý  - Xóa tất cả đơn hàng đang được xử lý đồng bộ.  - Lưu tất cả đơn hàng đã xóa vào bảng History  **broadcastOrders**  gửi xuống clients tất cả đơn hàng trống |
| **printHelper**  Được thực hiện khi dưới client có một trong số các hành động  -Báo bếp  -Thanh toán đơn hàng  và có thiết lập In nhờ  ( chỉ gửi lệnh in nhờ từ App )  Khi Client nhận được tín hiệu **printHelper**  sẽ gọi hàm In để in  ( chỉ nhận lệnh in trên desktop và clietn có thiết lập nhận lệnh in nhờ) | **printHelper**  Nếu ShiftId trùng với ShiftId hiện tại trên server tiến hành xử lý  forward tín hiệu từ client gửi lên xuống tất cả các client khác |
| **broadcastOrders**  Khi có 1 user gửi tín hiệu lên socket, server sẽ trả về cho tất cả User danh sách dữ liệu hiện tại trên server. ( trừ printHelper chỉ trả về đơn hàng cần in )  Khi client nhận được tín hiện **broadcastOrder**  -> Kiểm tra storeId trong data nhận được, nếu currentStore.storeID không trùng với storeId nhận được thì ko xử lý.  -> Kiểm tra shiftId nhận được với shiftId dưới localStorage, nếu shiftId không trùng với shiftId dưới localStorage thì xóa shiftId & danh sách đơn hàng lưu theo storeId dưới localStorage. Reload App.  ->Nếu shiftId trùng với shiftId dưới localStorage thì xử lý tiếp :  -> Lọc ra danh sách đơn hàng chưa thực hiện báo bếp (saleOrder.hasNotice = true)  -> Ghi đè dữ liệu từ socket gửi về vào localStorage  -> Ghi đè danh sách đơn hàng chưa báo bếp ( vừa lọc ở trên ) vào localStorage  -> Kiểm tra đơn hàng trong từng bàn, cập nhật tableStatus = 1 cho những bàn đã có đơn hàng. | **broadcastOrders**  gửi xuống clients tất cả đơn hàng đang có tại server |
| **exception** |  |
| **invalidShift**  Khi user gửi lên shiftId ko match với server, server sẽ trả về exception **invalidShift**  Client nhận được exception invalidShift sẽ thực hiện  - Xóa shiftId lưu dưới localStorage  - Reload App  **invalidStore/** **unauthorizedClientId**  Khi client gửi lên thông tin storeId / session đăng nhập user ko đúng ( hết hạn ) thì server sẽ trả về exception **invalidStore** hoặc **unauthorizedClientId**  Client nhận được exception  **invalidStore** hoặc **unauthorizedClientId** sẽ thực hiện logout, đăng nhập lại để lấy thông tin từ user đăng nhập |  |

**D. Mô tả chi tiết tính năng**

**a. Đăng nhập**

Khi submit đăng nhập, sẽ đi lấy tuần tự các thông tin

- Get user session : AuthUrl+'auth/hugate?format=json&'

- Get access token: AuthUrl+'provider/GetUserSession?format=json&',

- bootloader : ApiUrl+'bootloader?format=json',

- getStoreList : ApiUrl+'stores?format=json',

- authBootloader : AuthUrl+'bootloader?format=json',

Các thông tin này sẽ được lưu xuống localStorage, sau khi đăng nhập thành công sẽ chuyển user sang màn hình POS.

Khi user truy cập vào PM, sẽ kiểm tra các thông tin như trên có được lưu dưới localStorage hay ko, nếu ko có lưu sẽ redirect sang màn hình đăng nhập.

Khi user vào trang login ( http://cafe.suno.vn/login ), initLogin sẽ được gọi, nếu dưới localStorage có lưu key account, thì hệ thống sẽ tự động thực hiện đăng nhập sau 2 giây.

**b. Tạo sơ đồ phòng bàn**

Màn hình tạo sơ đồ phòng bàn sẽ hiển thị khi trên hệ thống chưa có sơ đồ phòng bàn và user đăng nhập có quyền chủ cửa hàng.

Nếu hệ thống chưa có sơ đồ phòng bàn, và user đăng nhập không phải là chủ cửa hàng thì hệ thống tự sinh ra 1 bàn mang về.

Dữ liệu sơ đồ phòng/bàn được lấy ở API Setting với key = tableSetting

ApiUrl+'setting/getKeyValue?format=json&key=**tableSetting**

Dữ liệu sơ đồ phòng/bàn được lưu ở API Setting với key=**tableSetting**

ApiUrl+'setting/postKeyValue?format=json',

**User nhập thông tin sơ đồ phòng/bàn , Khu vực**

-> Push thêm thông tin bàn Mang về vào danh sách bàn User đã nhập

-> Tạo tableUuid cho từng bàn

-> Lưu thiết lập sơ đồ qua API Setting

-> Gọi kết ca

**User cập nhật sơ đồ bàn, khu vực: thực hiện lại thao tác tạo mới sơ đồ**

-> Push thêm thông tin bàn Mang về vào danh sách bàn User đã nhập

-> Tạo tableUuid cho từng bàn

-> Lưu thiết lập qua API Setting

-> Gọi kết ca

**c. Tạo đơn hàng**

Khi thêm hàng hóa vào đơn hàng, kiểm tra Object saleOrder đã có **saleOrderUuid** hay chưa, nếu chưa có thì tạo saleOrderUuid. saleOrderUuid là mã đơn hàng sử dụng để xác định đơn hàng khi xử lý đồng bộ dữ liệu client-server

**Khi thêm hàng hóa vào đơn hàng :**

* Kiểm tra saleOrder.createdBy, nếu không trùng với User hiện tại và User hiện tại không phải từ cấp Quản lý trở lên thì hiện thông báo " Không được phép thao tác trên đơn hàng của nhân viên khác".
* Nếu hàng hóa thêm là hàng có Serial thì hiện thông báo cho User
* Nếu hàng có quản lý tồn kho & không cho bán âm & số lượng hiện tại <=0 thì hiện thông báo cho User
* Nếu hàng hóa thêm vào là hàng hóa đã cấu hình sử dụng tính giờ và hàng hóa này đã có trong đơn hàng thì không cho thêm vào đơn hàng và hiện thông báo cho User
* Nếu hệ thống **Không** cấu hình "Tách riêng từng món trong hóa đơn" (printSetting.unGroupItem = false hoặc null, mặc định là null) và :

Hàng hóa chưa có trong đơn hàng thì set

item.quantity = 1; ( Số lượng của item này trong đơn hàng )

item.newOrderCount = 1; ( Số lượng của item này chưa báo bếp trong đơn hàng )

Hàng hóa đã có trong đơn hàng thì

tăng item. quantity lên 1 đơn vị

tăng item.newOrderCount lên 1 đơn vị

**d. Đơn hàng cho quán trà sữa, đơn hàng dịch vụ tính giờ**

**Đơn hàng cho quán trà sữa**

Nếu hệ thống Có cấu hình "Tách riêng từng món trong hóa đơn"

(printSetting.unGroupItem = true)

Khi thêm hàng hóa vào đơn hàng:

* SET

item.quantity = 1;

item.newOrderCount = 1;

* Kiểm tra có hàng hóa nào đang được PIN hay không ( check trên màn hình đơn hàng ).

Nếu có hàng hóa đang được PIN thì hàng mới thêm vào sẽ được đánh dấu item.isChild = '(+)' và hàng hóa mới thêm vào sẽ được push vào vị trí sau hàng hóa được Pin trong array saleOrder. orderDetails.

Nếu không có hàng hóa được PIN thì hàng hóa mới thêm vào sẽ được push vào cuối array saleOrder. orderDetails

**Đơn hàng cho dịch vụ tính giờ**

Khi thêm hàng hóa vào đơn hàng

-> Nếu có sử dụng dịch vụ tính giờ & Không áp dụng cho tất cả hàng hóa & Hàng hóa vừa thêm vào nằm trong danh sách hàng hóa đã cấu hình sử dụng dịch vụ tính giờ

* SET item.isServiceItem = true (đánh dấu hàng hóa này là hàng hóa tính giờ).
* SET item.timer = true (item.timer Dùng để hiển thị các nút Play , Stop, set True cho biết đang tính giờ và hiển thị nút Stop)
* SET item.timeCounter = 0 (ghi nhận thời gian sử dụng dịch vụ, tính từ khi thêm hàng hóa vào đơn hàng đến khi bấm nút ngưng tính giờ, tính theo millisecond).
* SET item.startTime = thời gian hiện tại (ghi nhận giờ gian bắt đầu sử dụng- Giờ vào).

-> Nếu có sử dụng dịch vụ tính giờ & Áp dụng cho tất cả hàng hóa

Chỉ SET item.isServiceItem = true

Khi nhấn vào nút bắt đầu tính giờ trên 1 item ( nút Play ) mới

SET item.timeCounter = 0

SET item.startTime = thời gian hiện tại

Khi ngưng tính giờ 1 hàng hóa

SET

* item.endTime = thời điểm ngừng tính giờ
* item.timeCounter = thời gian tính từ lúc bắt đầu tính giờ đến lúc ngừng tính giờ
* item.duration = String (thời gian sử dụng dịch vụ - Dùng để hiển thị thời gian kết thúc sử dụng dịch vụ - Giờ ra)
* item.blockCount = String (thời gian sử dụng dịch vụ được làm tròn theo block - Dùng để in ra hóa đơn)
* item.quantity = Math.ceil(item.timeCounter/ (60000 \* $scope.blockCounter)) \* ($scope.blockCounter/60);

Mặt định 1 đơn vị của item tính giờ = 1 giờ. Khi ngừng tính giờ cho 1 item, sẽ thực hiện quy đổi item.timeCounter sang item.quantity và làm tròn theo block đã được thiết lập.

Gọi socket updateOrder, gửi dữ liệu đơn hàng đang thao tác lên server

**e. In bếp / in tem báo bếp cho trà sữa**

**In bếp**

Khi user nhấn nút Báo bếp trên màn hình Pos

-> Kiểm tra user có quyền thao tác trên đơn hàng hiện tại không

-> Nếu user có quyền thao tác thì kiểm tra đơn hàng hiện tại còn món chưa báo bếp không

-> Nếu còn món chưa báo bếp:

->Tách những món chưa báo bếp trong đơn hàng ra đơn hàng tạm (item. newOrderCount > 0 là những món chưa báo bếp)

-> Gọi lệnh in đơn hàng tạm

-> Lưu đơn hàng tạm vừa in vào danh sách những phiếu báo bếp đã in saleOrder.printed[].

-> reset item.newOrderCount = 0 cho những món vừa báo bếp

-> reset item.comment = '' cho những món vừa báo bếp (item.comment là ghi chú cho hàng hóa trong đơn hàng sử dụng khi in hóa đơn báo bếp)

-> Gọi socket updateOrder, gửi dữ liệu đơn hàng đang thao tác lên server

**In tem báo bếp**

Chỉ thực hiện được trên phiên bản web

Khi nhấn nút Báo bếp trên màn hình Pos

Nếu hệ thống có thiết lập **"Tách riêng từng món trong hóa đơn"** và **"In tem nhãn khi báo bếp"**

**->** Thực hiện xử lý đơn hàng trước khi gửi lệnh in

Kiểm tra item trong đơn hàng

->Xác định những item có tồn tại item.isChild ( là những món phụ )

->Push tên những item phụ vào item.childItem của item chính trước nó (ko tồn tại item.isChild).

item.childItem được sử dụng để in tên món phụ trong tem báo bếp.

->Xóa những item có item.isChild ra khỏi đơn hàng.

Trả về danh sách đơn hàng đã xử lý và gửi lệnh in hóa đơn.

**f. In Bar / bếp**

Cho phép user thiết lập danh sách những hàng hóa thuộc quầy Bar khi cấu hình **"Tách riêng hóa đơn cho bar & bếp"** được bật

Danh sách hàng hóa thuộc quầy bar được lấy ở API Setting với key = BarItemSetting

ApiUrl+'setting/getKeyValue?format=json&key=**BarItemSetting**

Danh sách hàng hóa thuộc quầy bar được lưu ở API Setting với key=**BarItemSetting**

ApiUrl+'setting/postKeyValue?format=json'

Khi in hóa đơn

->Kiểm tra và tách hóa đơn cho những item có trong danh sách **BarItemSetting.**

Gọi lệnh in 2 lần, 1 lần cho danh sách có trong **BarItemSetting** vừa tách**,** 1 lần cho danh sách item còn lại trong đơn hàng.

**g. Kết ca**

Chỉ được thực hiện khi user có quyền quản lý trở lên.

->Client gửi tín hiệu **completeShift** lên server

->Xóa danh sách đơn hàng lưu theo StoreId dưới localStorage

->Cập nhật tồn quỹ đầu ngày về 0 ở API Setting với key : getBalance={storeId}

-> Reload app