

**COMMIT CODE TO SVN EVERY END OF DAY**

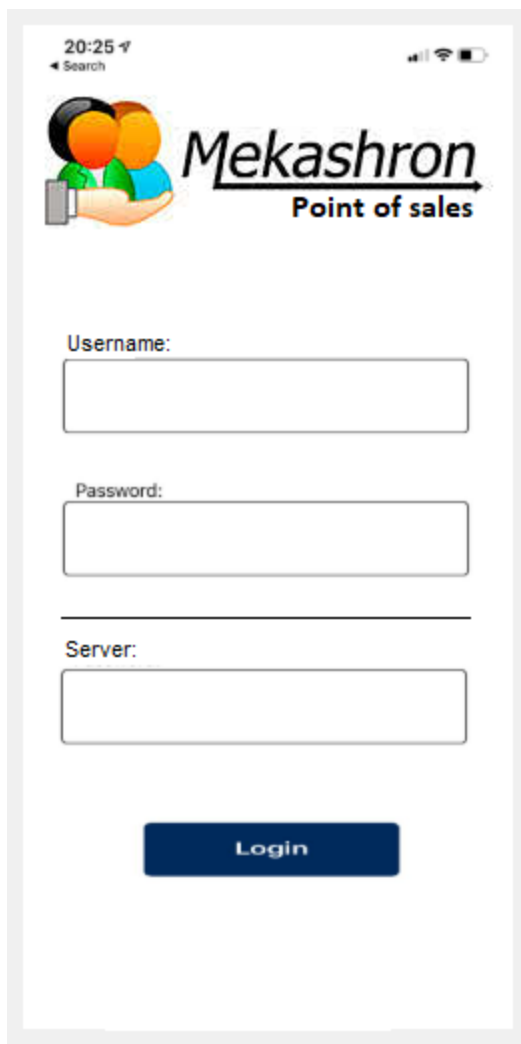
Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.

For any questions I am available by skype [yan@mekashron.com](mailto:yan@mekashron.com)


Example web service (ISAPI) is running on <http://isapi.mekashron.com:33322/>

POS should work with local tinySQL DB and update it's DB when internet connection is available update from web service local DB.

- Login screen:  
You need to call %server%.ol\_login function. If return success continue to the next screen. In case of an error show the error.



20:25 ← Search



Username:

Password:

---

Server:

**Login**

After login, store the credentials and next app start skip this screen.

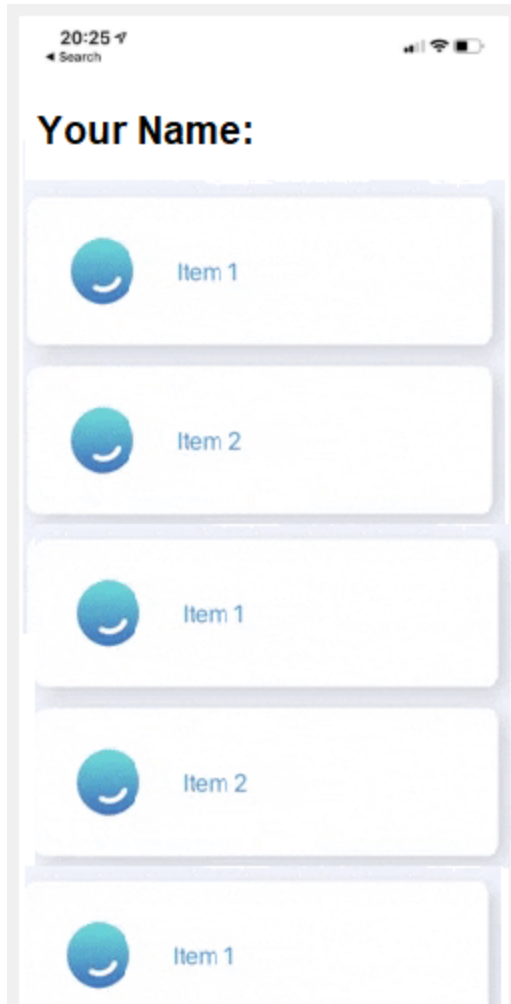
- If ol\_login function return you that logged entity is isBusiness=true.

**COMMIT CODE TO SVN EVERY END OF DAY**

Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.

Call function GetUsers. And show list of allowed users to use the POS. recommended UI components:

<https://github.com/Trixieapp/react-virgin>



Else if entity is isUser=true call the function GetBusiness, if list return more than 1 business show similar list with title. **Your business:**

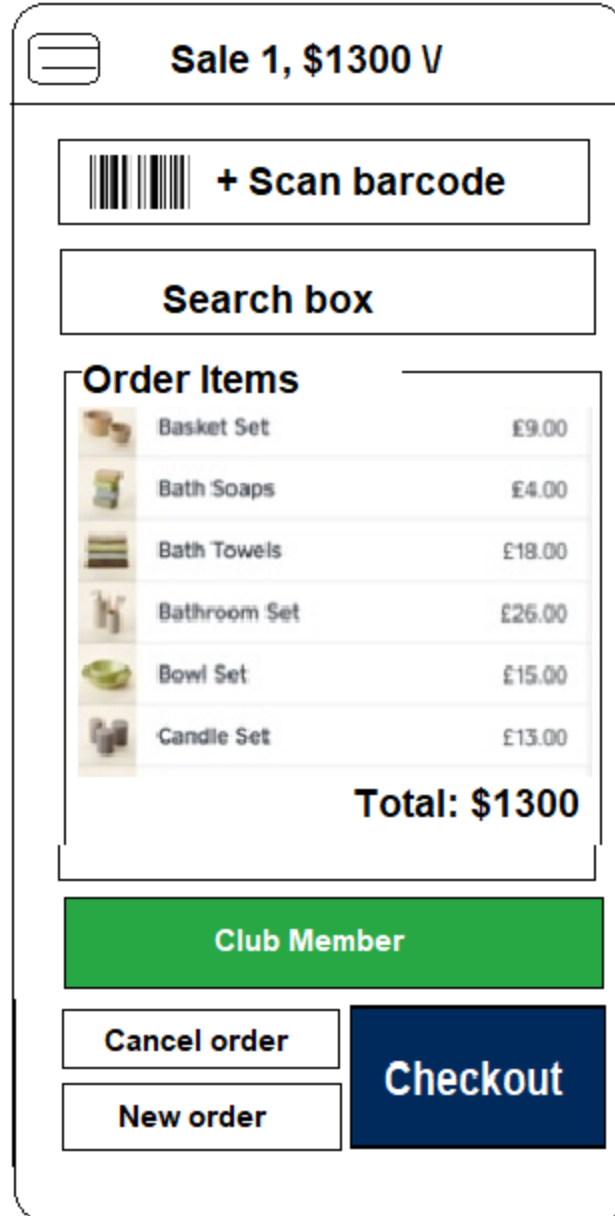
- Next screen show order option. Use function GetInventory() to preload all Inventory into the DB.
  - \* Clicking on barcode will open option to scan barcode using camera and search for matching product in local DB. if found add to order.
  - \* Search, will allow user to type any product and find it from the list of products, clicking on a product will add to the list.
  - \* Clicking on a product in the list will open new screen with option to edit qty, discount and delete the product from order at all.

**COMMIT CODE TO SVN EVERY END OF DAY**

Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.

**Menu options**

Refunds  
Transactions  
Settings  
Logout



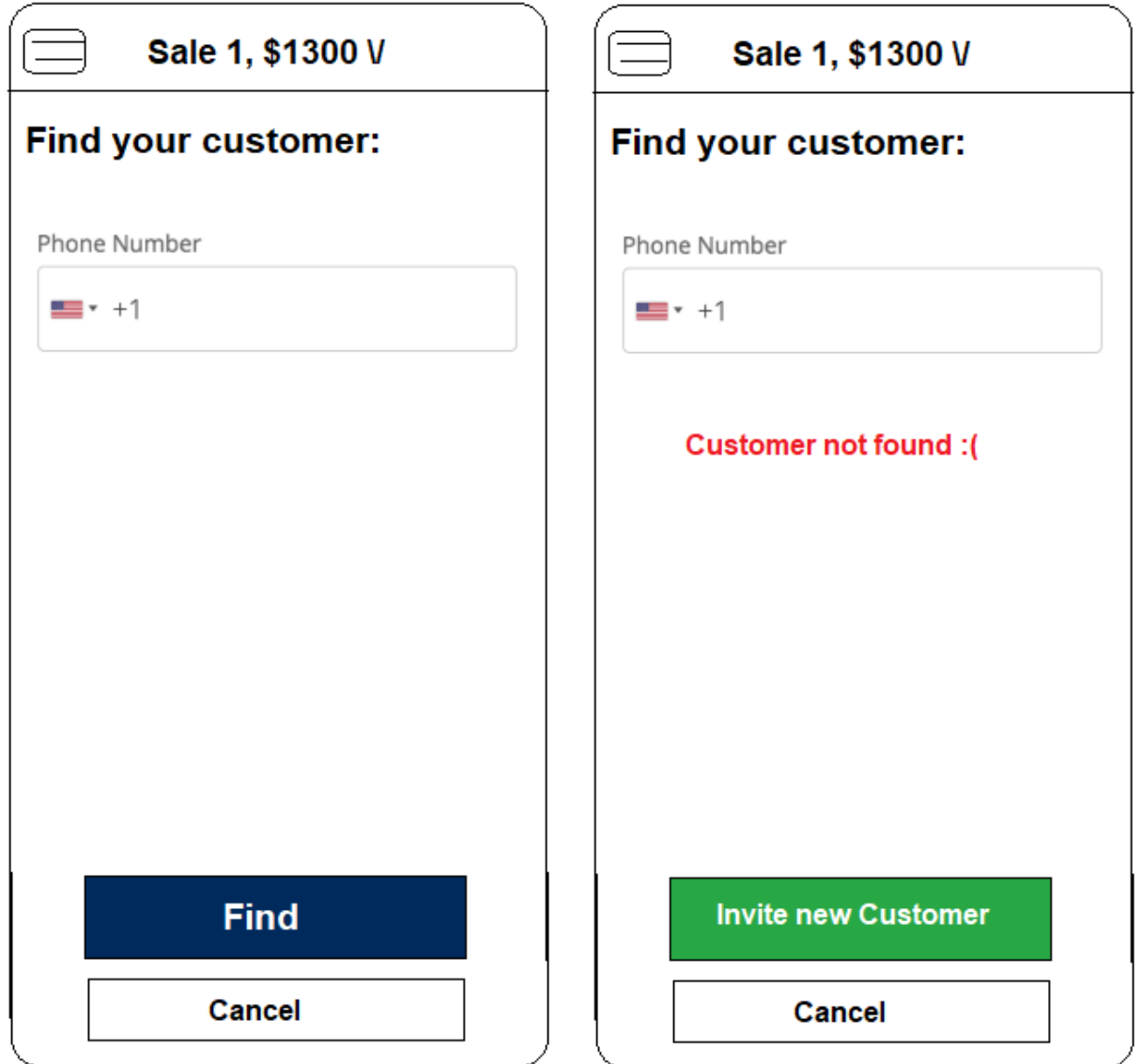
\* cancel order, will clear current order.

\* New order will add order and user can select between orders by clicking on header.

- Clicking on club member button

**COMMIT CODE TO SVN EVERY END OF DAY**

Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.

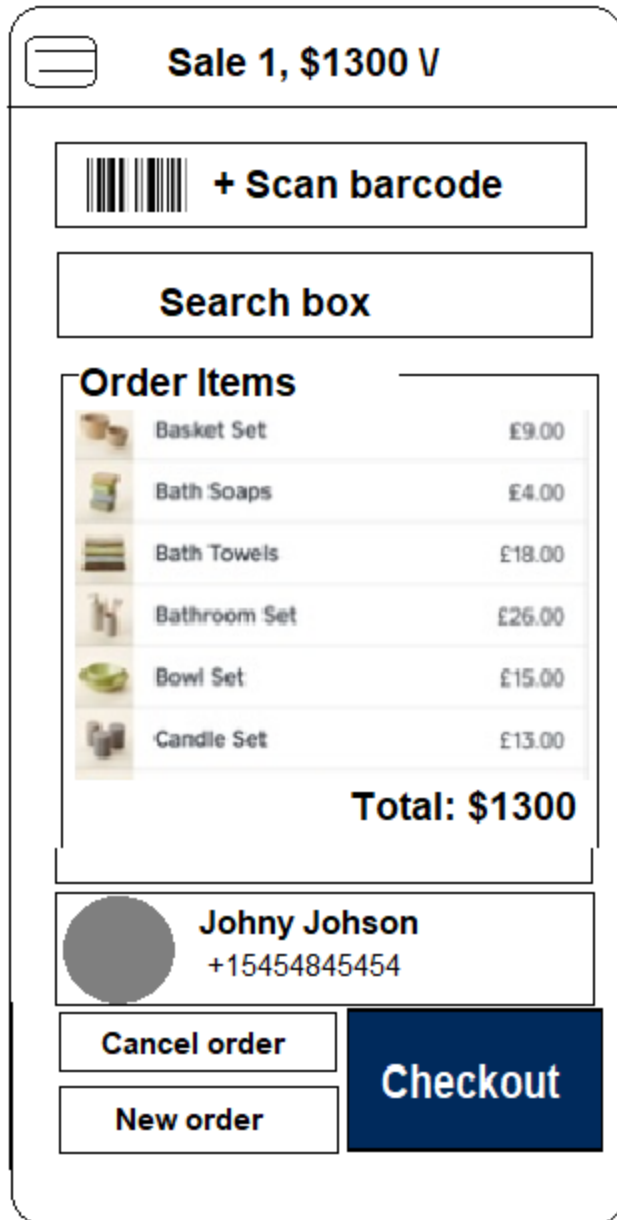


To find a customer call the function FindEntity(entity login, phone, %phone num%).  
If not found,

**COMMIT CODE TO SVN EVERY END OF DAY**

Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.

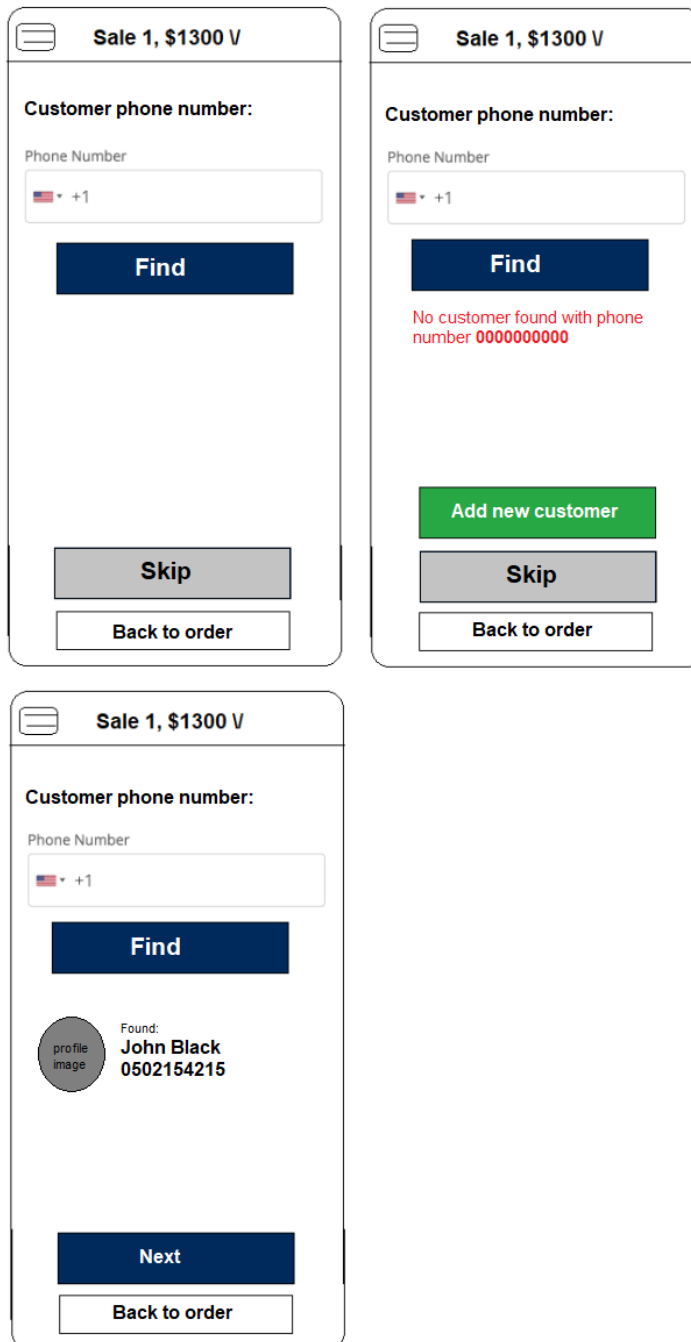
If found, return to order screen like that:



**COMMIT CODE TO SVN EVERY END OF DAY**

Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.

- Clicking on Checkout, next screen if member not assigned to order yet. Show screen where user can assign customer to the order.



- Next step, call GetPaymentsMethod store the payment methods in local DB. to get list of available payment methods show them in list. When clicked on payment method.

*The payment method list use normal large font. Ignore example image small font.*

**COMMIT CODE TO SVN EVERY END OF DAY**

Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.

The screenshot shows a mobile application interface for a sale. At the top, there is a hamburger menu icon and the text "Sale 1, \$1300 V". Below this, a circular profile image placeholder is labeled "profile image". To its right, the text "Customer:" is followed by "John Black" and "0502154215". The amount "R2.58" is displayed prominently in the center, with the instruction "Choose a payment type below" underneath. There are five payment options listed, each with a right-pointing chevron: "Cash", "Record Card Payment", "Cash", "Record Card Payment", and "Cash". At the bottom of the screen is a button labeled "Back to order".

Show option to type how much user wants to pay using this method. If user type more than the remaining amount, change the remaining amount to show as Change:

**COMMIT CODE TO SVN EVERY END OF DAY**

Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.


The screenshot shows a mobile application interface for a point-of-sale system. At the top, it displays 'Sale 1, \$1300 V'. Below this, there is a customer profile section with a circular placeholder for a 'profile image' and the text 'Customer: John Black 0502154215'. The main section is titled 'Cash' and shows a 'Remanding:' field with the value '\$1.58'. Below this, there is a prompt 'or type amount' and another input field containing '\$1.58'. At the bottom, there is a numeric keypad with digits 1-9, 0, and a decimal point, along with a blue 'enter' button and a backspace icon.


If user type amount less than remanding return back to payment method selection and show using what payment part of the order already paid.



**COMMIT CODE TO SVN EVERY END OF DAY**

Completed task mark **GREEN**. Task marked **RED** are not working and needs to be fixed.

 **Sale 1, \$1300 V**

 Customer:  
**John Black**  
**0502154215**

**₪2.58**  
Choose a payment type below

**( ) \$1.00 Cash** [Delete](#)

---

Cash >

---

Record Card Payment >

---

Cash >

---

Record Card Payment >

**Back to order**