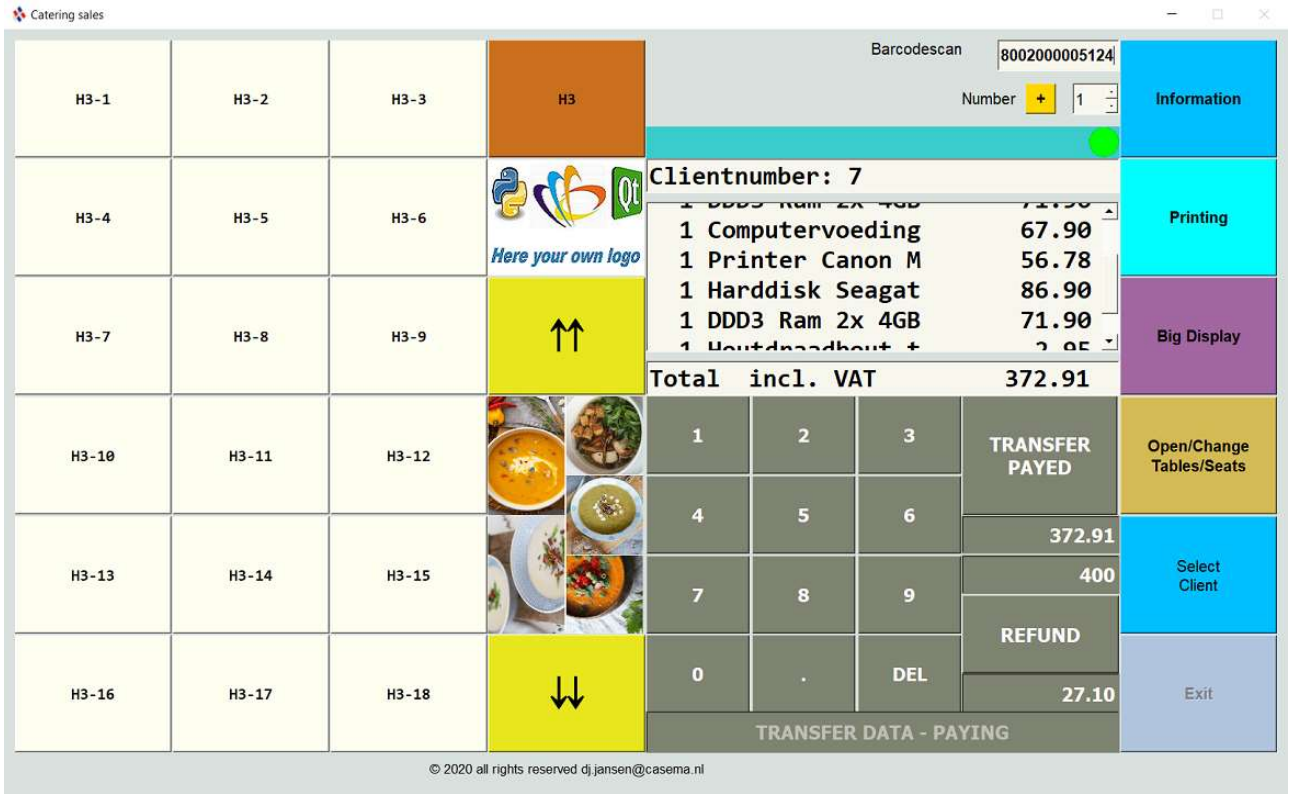
Documentation Catering Point Of Sale. (POS)



Screenshot mainscreen.

Database structure catering:

```
catering=# \d
              List of relations
 Schema |         Name          | Type  |  Owner
--------+-----------------------+-------+----------
 public | additional            | table | postgres
 public | article_grouplines    | table | postgres
 public | articles              | table | postgres
 public | buttons               | table | postgres
 public | clients               | table | postgres
 public | employees             | table | postgres
 public | groupbuttons          | table | postgres
 public | invoices              | table | postgres
 public | loss                  | table | postgres
 public | order_lines           | table | postgres
 public | params                | table | postgres
 public | payments              | table | postgres
 public | purchase_orderlines   | table | postgres
 public | sales                 | table | postgres
 public | suppliers             | table | postgres
 public | tables_layout         | table | postgres
```

```
catering=# \d additional
                              Table "public.additional"
      Column       |         Type          | Collation | Nullable |        Default
-------------------+-----------------------+-----------+----------+-----------------------
 addID             | integer               |           | not null |
 barcode           | character varying(13) |           |          | ''::character varying
 description       | character varying(13) |           |          | ''::character varying
 item_price        | double precision      |           |          | 0
 number            | double precision      |           |          | 0
 item_unit         | character varying(16) |           |          | ''::character varying
 article_group     | character varying(40) |           |          | ''::character varying
 location_warehouse | character varying(8)  |           |          | ''::character varying
Indexes:
    "additional_pkey" PRIMARY KEY, btree ("addID")
```

```
catering=# \d article_grouplines
                         Table "public.article_grouplines"
    Column     |         Type          | Collation | Nullable |        Default
---------------+-----------------------+-----------+----------+-----------------------
 lineID        | integer               |           | not null | 0
 grouplinetext | character varying(40) |           |          | ''::character varying
Indexes:
    "article_grouplines_pkey" PRIMARY KEY, btree ("lineID")
```

```
catering=# \d articles
                                  Table "public.articles"
       Column        |         Type          | Collation | Nullable |          Default
---------------------+-----------------------+-----------+----------+---------------------------
 barcode             | character varying(13) |           | not null |
 description         | character varying(50) |           |          | ''::character varying
 item_price          | double precision      |           |          | 0
 item_stock          | double precision      |           |          | 0
 item_unit           | character varying(16) |           |          | ''::character varying
 minimum_stock       | double precision      |           |          | 0
 order_size          | double precision      |           |          | 0
 location_warehouse  | character varying(8)  |           |          | ''::character varying
 article_group       | character varying(40) |           |          | ''::character varying
 thumbnail           | character varying(50) |           |          | ''::character varying
 category            | integer               |           |          | 0
 order_balance       | double precision      |           |          | 0
 order_status        | boolean               |           |          | true
 mutation_date       | character varying(10) |           |          | ''::character varying
 annual_consumption_1 | double precision      |           |          | 0
 annual_consumption_2 | double precision      |           |          | 0
 VAT                 | character varying(4)  |           |          | 'high'::character varying
 short_descr         | character varying(15) |           |          | ''::character varying
 selling_price       | double precision      |           |          | 0
 selling_contents    | double precision      |           |          | 0
 additional          | integer               |           |          | 0
 supplierID          | integer               |           |          | 0
 ordering_manual     | integer               |           |          | 0
Indexes:
    "barcode_pkey" PRIMARY KEY, btree (barcode)
    "barcode_idx" btree (barcode)
Referenced by:
    TABLE "loss" CONSTRAINT "barcode_fkey" FOREIGN KEY (barcode) REFERENCES articles(barcode)
```

```
catering=# \d buttons
                                  Table "public.buttons"
   Column   |         Type          | Collation | Nullable |           Default
------------+-----------------------+-----------+----------+-----------------------------
 buttonID   | integer               |           | not null |
 buttontext | character varying(60) |           |          | ''::character varying
 barcode    | character varying(13) |           | not null | ''::character varying
 reference  | character varying(5)  |           |          | ''::character varying
 accent     | integer               |           |          | 0
 bg_color   | character varying(7)  |           |          | '#FFFFF0'::character varying
Indexes:
    "buttonID_pkey" PRIMARY KEY, btree ("buttonID")
```

```
catering=# \d clients
                        Table "public.clients"
  Column  |         Type          | Collation | Nullable |       Default
----------+-----------------------+-----------+----------+----------------------
 clientID | integer               |           | not null |
 employee | character varying(20) |           |          | ''::character varying
 barcode  | character varying(8)  |           |          | ''::character varying
Indexes:
    "clients_pkey" PRIMARY KEY, btree ("clientID")
Referenced by:
    TABLE "order_lines" CONSTRAINT "clientID_fkey" FOREIGN KEY ("clientID") REFERENCES clients("clientID")
```

```
catering=# \d employees
                             Table "public.employees"
  Column   |         Type          | Collation | Nullable |        Default
-----------+-----------------------+-----------+----------+-----------------------
 barcodeID | character varying(8)  |           | not null |
 firstname | character varying(20) |           |          | ''::character varying
 lastname  | character varying(30) |           |          | ''::character varying
 access    | integer               |           |          | 1
 callname  | character varying(20) |           |          | ''::character varying
Indexes:
    "barcodeID_pkey" PRIMARY KEY, btree ("barcodeID")
```

```
catering=# \d groupbuttons
                                Table "public.groupbuttons"
     Column      |         Type          | Collation | Nullable |           Default
-----------------+-----------------------+-----------+----------+-----------------------------
 groupID         | integer               |           | not null | 0
 reference       | character varying(2)  |           |          | ''::character varying
 buttongrouptext | character varying(60) |           |          | ''::character varying
 bg_color        | character varying(7)  |           |          | '#FFFFF0'::character varying
Indexes:
    "groupbuttons_pkey" PRIMARY KEY, btree ("groupID")
```

```
catering=# \d invoices
                            Table "public.invoices"
   Column    |         Type          | Collation | Nullable |        Default
-------------+-----------------------+-----------+----------+-----------------------
 invoiceID   | integer               |           | not null |
 barcode     | character varying(13) |           |          | ''::character varying
 description | character varying(50) |           |          | ''::character varying
 delivery    | double precision      |           |          | 0
 item_price  | double precision      |           |          | 0
 supplierID  | integer               |           |          | 0
 orderlineID | integer               |           |          | 0
 paydate     | character varying(10) |           |          | ''::character varying
 bookdate    | character varying(10) |           |          | ''::character varying
 item_unit   | character varying(16) |           |          | ''::character varying
Indexes:
    "invoices_pkey" PRIMARY KEY, btree ("invoiceID")
```

```
catering=# \d loss
                          Table "public.loss"
  Column  |         Type          | Collation | Nullable |        Default
----------+-----------------------+-----------+----------+-----------------------
 lossID   | integer               |           | not null |
 number   | double precision      |           |          | 0
 category | character varying(22) |           |          | ''::character varying
 bookdate | character varying(10) |           |          |
 barcode  | character varying(13) |           |          |
Indexes:
    "lossID_pkey" PRIMARY KEY, btree ("lossID")
    "fki_barcode_fkey" btree (barcode)
Foreign-key constraints:
    "barcode_fkey" FOREIGN KEY (barcode) REFERENCES articles(barcode)
```

```
catering=# \d order_lines
                          Table "public.order_lines"
    Column     |         Type          | Collation | Nullable |        Default
---------------+-----------------------+-----------+----------+-----------------------
 ID            | integer               |           | not null |
 barcode       | character varying(13) |           |          | ''::character varying
 description   | character varying(40) |           |          | ''::character varying
 short_descr   | character varying(15) |           |          | ''::character varying
 number        | double precision      |           |          | 0
 item_price    | double precision      |           |          | 0
 sub_total     | double precision      |           |          | 0
 sub_vat       | double precision      |           |          | 0
 mutation_date | character varying(10) |           |          | ''::character varying
 callname      | character varying(20) |           |          | ''::character varying
 clientID      | integer               |           |          | 0
Indexes:
    "tables_accounts_pkey" PRIMARY KEY, btree ("ID")
    "clientID_fkey" btree ("clientID")
```

```
catering=# \d params
                          Table "public.params"
 Column  |         Type          | Collation | Nullable |        Default
---------+-----------------------+-----------+----------+-----------------------
 paramID | integer               |           | not null |
 item    | character varying(20) |           |          | ''::character varying
 value   | double precision      |           |          | 0
Indexes:
    "paramID_pkey" PRIMARY KEY, btree ("paramID")
```

```
catering=# \d payments
                          Table "public.payments"
    Column     |         Type          | Collation | Nullable |        Default
---------------+-----------------------+-----------+----------+-----------------------
 payID         | integer               |           | not null |
 kind          | character varying(25) |           |          | ''::character varying
 amount        | double precision      |           |          | 0
 bookdate      | character varying(10) |           |          | ''::character varying
 paydate       | character varying(10) |           |          | ''::character varying
 instance      | character varying(25) |           |          | ''::character varying
 accountnumber | character varying(25) |           |          | ''::character varying
 ovorderID     | integer               |           |          | 0
 basis         | double precision      |           |          | 0
Indexes:
    "payID_pkey" PRIMARY KEY, btree ("payID")
```

```
catering=# \d purchase_orderlines
                       Table "public.purchase_orderlines"
     Column      |         Type          | Collation | Nullable |        Default
-----------------+-----------------------+-----------+----------+-----------------------
 orderlineID     | integer               |           | not null |
 barcode         | character varying(13) |           |          | ''::character varying
 description     | character varying(50) |           |          | ''::character varying
 item_price      | double precision      |           |          | 0
 item_unit       | character varying(16) |           |          | ''::character varying
 item_stock      | double precision      |           |          | 0
 minimum_stock   | double precision      |           |          | 0
 order_size      | double precision      |           |          | 0
 ordering_manual | integer               |           |          | 0
 supplierID      | integer               |           |          | 0
 bookdate        | character varying(10) |           |          | ''::character varying
 ordered         | double precision      |           |          | 0
 order_date      | character varying(10) |           |          | ''::character varying
 delivery        | double precision      |           |          | 0
 delivery_date   | character varying(10) |           |          | ''::character varying
Indexes:
    "purchase_orderlines_pkey" PRIMARY KEY, btree ("orderlineID")
```

```
catering=# \d sales
                              Table "public.sales"
    Column     |         Type          | Collation | Nullable |        Default
---------------+-----------------------+-----------+----------+-----------------------
 salesID       | integer               |           | not null |
 receiptnumber | integer               |           |          | 0
 barcode       | character varying(13) |           |          | ''::character varying
 description   | character varying(40) |           |          | ''::character varying
 number        | double precision      |           |          | 0
 item_price    | double precision      |           |          | 0
 sub_total     | double precision      |           |          | 0
 sub_vat       | double precision      |           |          | 0
 callname      | character varying(20) |           |          | ''::character varying
 mutation_date | character varying(10) |           |          | ''::character varying
 short_descr   | character varying(15) |           |          | ''::character varying
 clientID      | integer               |           |          | 0
Indexes:
    "ID_pkey" PRIMARY KEY, btree ("salesID")
```

```
catering=# \d suppliers
                            Table "public.suppliers"
    Column    |          Type           | Collation | Nullable |        Default
--------------+-------------------------+-----------+----------+-----------------------
 supplierID   | integer                 |           | not null | 0
 company_name | character varying(40)   |           |          | ''::character varying
 street       | character varying(40)   |           |          | ''::character varying
 housenumber  | character varying(14)   |           |          | ''::character varying
 residence    | character varying(40)   |           |          | ''::character varying
 telephone    | character varying(13)   |           |          | ''::character varying
 email        | character varying(200)  |           |          | ''::character varying
 addition     | character varying(1000) |           |          | ''::character varying
 zipcode      | character varying(7)    |           |          | ''::character varying
Indexes:
    "suppliers_pkey" PRIMARY KEY, btree ("supplierID")
```

```
catering=# \d tables_layout
                         Table "public.tables_layout"
   Column   |         Type          | Collation | Nullable |        Default
------------+-----------------------+-----------+----------+-----------------------
 ID         | integer               |           | not null |
 occupied   | integer               |           |          | 0
 table_seat | character varying(4)  |           |          | ''::character varying
 clientID   | integer               |           |          | 0
 callname   | character varying(20) |           |          | ''::character varying
Indexes:
    "tables_layout_pkey" PRIMARY KEY, btree ("ID")
    "fki_clientID_fkey" btree ("clientID")
```

Point of Sale Catering

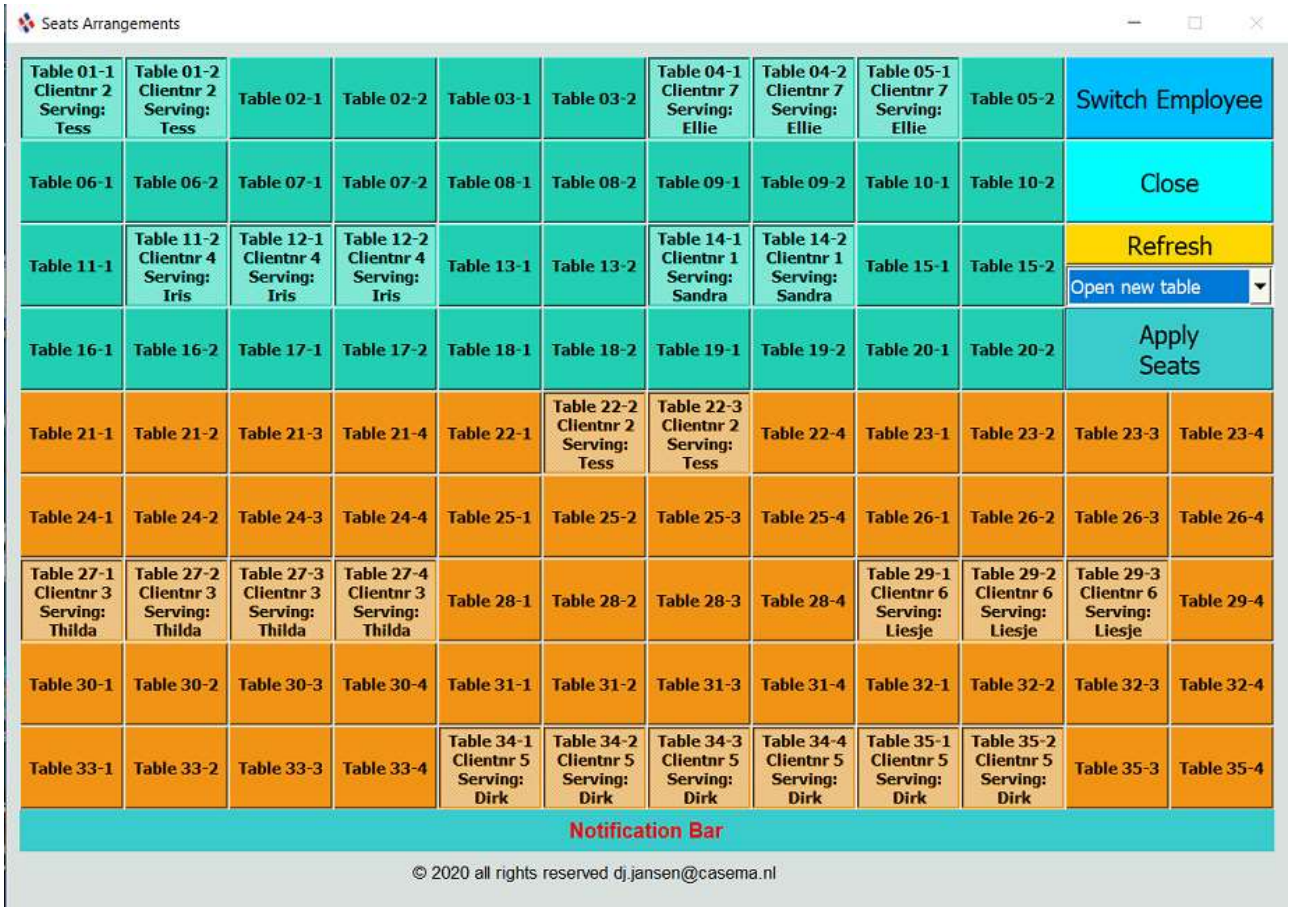
Custom daily operations

The system detects if a logon barcode or a product barcode is scanned.
When no logon is established processing is blocked.

The red sign in the notification bar above the display is showed.
By valid logon the notification bar shows a green sign.
The logon is invisible with scanning.

When the logged employee logs the barcode a second time, the employee is logged out.
When another employee logs his barcode, the logon is switched towards this employee.

The normal operation starts with logon by barcode.
When customers arrive, the employee push the button “Open Table/Change Tables/Seats”
The table management screen is displayed:



Each button on the screen represents a seat. The green buttons represents tables paired with two seats. The brown buttons represents tables with 4 seats. In total 100 seats are available.
By clicking the buttons the seats are checked as occupied. For groups every combination of tables and seats is possible. For conformation chose “Open new table” (default choice) and push button “Apply Seats”. With this “Open new table” more arrangements scan be made after each other.
Chose the occupied seats and push the button “Apply seats” and repeat this for the next client or group of clients.
With applying a popup is showed, asking to print a client barcode. This serves the purpose to switch fast between clients with the printed barcode.
During the stay, it’s possible to add or to free up tables or seats by leaving or arriving members of the group. When no costs are made it it’s possible to remove all seats.
This is established with menu choice “Change table client <number> , with the number of the client who booked the arrangement. Only customers linked by the concerned employee are showed in the choice fields. For every client combination from the concerning employee is a line added in the menu choices.

When costs are made the last seat cannot be removed in this screen. It will be removed if the payment is done in the mainscreen.
On the buttons for the seats the table and seatnumber, clientnumber and serving employee is showed.
The buttons with there belongings and states are visible for every employee, but only operable for the employee who established the linking.

With the “Switch Employee” button it is possible to transfer the service to another employee by end of working time or interim absence.

If al buttons for the seats are pushed, push the “Refresh” button to make the changes visible, and push “Close” button to leave the tablemanagement screen. Next push the “Select Client” button to select the customers to serve, or for fast selecting or switching between clients use the barcode for the concerning client. Here only the linked clients for the concerning employee can be chosen.

After this steps has been accomplished, it it possible to choose products with the productbuttons, or scaning products. Default a number of 1 is chosen. If more of the same products is required, use the little spinbox to change the number (1 towards 99) before scanning of pushing a button. After scanning of each product or pushing a product button the number of the spinbox is reset to 1.
For return bookings a little button before the number is provided. Push this checkable button before operating and the numbers will be operable from -1 towards -99. After each operation the spinbox will been reset to it’s oriniginal state, to avoid wrong bookings.

The product buttons are organized as follows:
The 10 maingroups with the double arrows are accessable by pushing the arrowbuttons up or down. Here a picture of the maingroup is presented between the arrowbuttons. With the button above the logo 5 subgroups of each maingroup can be chosen. Next a productbutton can be pushed, possibly preceded with the right number if more then 1 product required. The consumptions and their prices are showed in the little display. Also the total amount of the consumptions is showed. For a clearer total overview push the button “Big Display”
After serving the client, the next client can be chosen, with the barcode, or by the button.
If the client belongs to another employee, first switch the employee with the barcode and next fulfill the client code scan or choose with button “Select client”

Booking additional products:
With selected client it is possible to book additional products. These products (for instance sugar, milkcups, potatoes,vegetables, meat and so on) are not direct attributable to a customer but booked on groupitems as additional costs. This transactions must be established if the products are picked from the storage.

If a client will pay the bill, then chose the concerning client, prints the receipt if desired, with the “Printing” button, by cash paying use the modest calculator on the screen to fill in the cash payed by the customer in the CASH field and push the “REFUND” button. The change will be calculated and showed in the CHANGE field, taking into account the rounding, which is set in advance. The customer also can pay by a pin transaction.
After payment is done press the “TRANSFER PAYED” button to make the necessary bookings and free the tables / seats for this customer. If no costs are made, it’s also possible to make a “null” booking to free tables and seats on the main screen.

A notification bar is visible for displaying several notifications during operation.

The following documentation concerns description of the backend for administration reasons and is accessable with level 2 by pushing the “Administration” button.

The structure of the adminmenu:

Employees submenu	*New employee* *View / change employee*
Articles submenu	*Insert new articles* *View / change articles* *Booking loss articles* *View loss articles*
Article-groups lines – view / change	
Purchase submenu	*Collecting purchases* *Manual ordering* *Unknown supplier ordering* *Receiving / processing deliveries* *Ordering suppliernumber suppliername* *Ordering suppl……* *…….* *……* *To order yet - view* *Ordered – view* *All orders – view* *Delivered orders - view*
Suppliers submenu	*Suppliers new* *Suppliers view / change* *Invoices suppliers /paying*
Imports submenu	*Import new articles* *Import price changes* *Import expired articles* *Import deliveries articles*
Groupbuttons grouptext /color	
Payments view / pay	
Parameters view / change	
Sales – view	
Additionals view	
Turnover submenu – view	*Daily gross turnover* *Monthly gross turnover* *Yearly gross turnover*
Reprint purchase orders	

Reprint sales receipts

We will describe the menuitems with their purpose and possibilities.

Employee submenu:

With the logon from barcode accesslevel 2 the “Administration” button is activated. With this button and submenu accounts it is possible to generate other barcodes for logon purposes.
The barcodefield will be generated as 5 random digits by the program. (The first 2 digits starts with 24 the 8th digit is a check number). The program will check and correct for duplicates. Access is default set on level 1, change if desired. If accesslevel is set to 0 processing is blocked for the employee.
The first name, last name and callname of the employee must be inserted.
The callname field will be printed on the saleslip, it’s also saved in the sales table.

The barcodelabel is saved in folder . /Barcodes/Employees/ after inserting a new employee, a barcode can be printed for logon in the menuchoice View / change employee with the print button. Before printing remove the number under barcode for security reasons and replace by logon name. (For instance use application Paint)
By the menuchoice “View / change employee” click the first field of the list with employees to select.

Articles submenu:
General remarks:
It is important to pay particular attention to the completion of the fields, for reasons of clarity and stock control.
Always fill in description (for purchase) and short description (for receipts and display text),
And surely in the start-up phase the order-size and minimum stock. When the system builds a history of usage, the system calculate the items minimum stock and order-size
In the table articles, the columns minimum stock, category, order_size, order_status, order_balance, annual_consumption1 and annual-consumption2 are present.
We will explain the purpose of these columns.
The minimum_stock column is used for purchasing, if item_stock + order_balance < minimum_stock a orderline is generated. These lines are generated in the “Purchase submenu” Collecting purchases.
If the purchase orderlines are collected the field order_status is set to False, so no further orders are generated until the products are delivered. The order-balance is icreased with the ordered number. After delivery the field order_status is reset to True, so the product is released for calculations of stock, order_balance is reduced with the delivered number.
The column category determine the delivery time. The category runs from 3 days to 1 year in 8 steps and 1 category daily fresh products. The latter is outside the supply system and must be manual handled.
The calculation of this items is established by the Camp formula:

Camp formula: $Q = \sqrt{2DF/HP}$
Stands for:
Q = Quantity

- D = Demand / year (table articles)
- F = Fixed Costs (order costs conversion costs) (from table params)
- H = Stock costs as a percentage of the price (from table params)

- P = Price of the product (from table articles)

The annual_consumption is counted by the article usage in a year.
Even year annual_consumption_1
Odd year annual_consumption_2
By starting of a new year the oldest year of annual consumption is set to 0, so the counting restarts, with this column, so always the last year is preserved for calculation of stock.

It is possible to influence the minimum stock and order size by changing the paramID 5 and 6 (Ordercosts and surcharge for interest and storage)

Columns description, short description, selling_price and selling_contents and their usage:

The column description is used for the product as purchased. For instance a bottle Chardonnay white wine 2007 of 0.75 liter. The short description can be in this case Glass white wine.
The item_price Euro 6.95. The selling_price 5.00 Euro (1 glass). The selling_price for the bottle 21.95. The selling_contents is in this case 0.15 for the glass, and for the bottle 1. The selling by glass can be established with a button coupled to barcodenumber and with 0.1 as selling_content.
By selling a bottle the item can be scanned or selled by button or both.

Articles:
By inserting articles 3 items are important. The checkbox manual , the checkbox additional and inserting articles inserting generated or scanned.
The checkbox manual will be set, if the manager wants to control the ordersize. This is surely in the startup phase important. The orderlines will be generated, but the order-size can be adjusted.
The checkbox additional must be set, when the product are not directy assignable and to calculate to the customer. For instance cleaning products, sugar, milkcups, cookies, potatoes, vegetables, meat and so on. These product will be booked on the several article_groups for further calculation.
By inserting articles it is possible to use a generated article by the system, or to use a scanned article product in the commercial range. The generated article is a EAN13 number starting with 28 (free, so no conflicts arise with commercial numbers), then 10 digits serial number and last a checksum number as a valitidy check. (usage for instance own products like maindish, desert, cup of coffee)
When a commercial number is wanted the generated number can be overruled in the last field.
Fill in all items, position the cursor in the last field and scan the product. The product is inserted without pushing the button, with the scanned code.
The other menu choices are for changing articles, booking loss articles, and view loss articles.
Loss articles are divided in obsolete, warehouse differences, damaged or shelf life.

Article-groups lines – view / change

The article-group lines are group classification of products.
The group-lines are to determine by the manager, because it integrated in the database.
Until 19 lines filled lines will be used by the program.
Click on the first field of the line from the list and change the contents if desired.

Purchase submenu

Collecting orderlines.
Showed as second line in the menu, because first line will be used more frequently.
This menuchoice iterates the whole table articles and generate on contents of the fields item_stock, minimum_stock, order_balance, order_status and category the orderlines.

The list is stored in the table purchase_orderlines, with the clientnr if known and also indicated or the orderline should be ordered manual.

Ordering / process deliveries / views

Suppliers

Imports

Groupbuttons and buttons