

## **1 Application Overview**

- The application must have proper validation, which tells the uses when they have not entered information in required fields and when they enter data in the wrong format (for example dates, monetary values and the MaxLength of text boxes).
- It must use appropriate icons.
- It must be developed using Visual Studio and SQL SERVER.
- In the database tables, the identification codes of each of the entities WILL NOT BE ENTERED BY THE USER.
- The data in the databases should be encrypted as far as possible. If the data can be read from the database engine it is wrong. It is understood that integers, chars, boolean and other types of data are not encrypted by convention. Card numbers must be encrypted even if the data is just numbers.

## **2 Project Description**

- The objective of the application is for it to serve for the management of internet food sales for a restaurant.
- In the app, there are two modules or clients (frontends). The first one will be an intranet client where the administrators of the app can manage the products they want to sell online.
- The second client is for the customers. This one is intended as a shop. It consists of an internet application where the restaurant's customers can search products and buy them by paying using a number of methods.
- A credit or debit card interface and an electronic check service (something like paypal) should also be developed so that customers can cancel their orders. The app will have a web service with a second database that the app will consume through the web service. The idea is that with this second database, the app can simulate consuming a web service that has its own database which will help the app with the payment process.

## **3 The Client for the Administrators**

- The scope of the first, administrative module is as follows:
  1. Registration and updating of general use catalogs
    - i. Users
    - ii. Roles
    - iii. Consecutives
    - iv. Types of Prices
    - v. Card Types

- vi. Payment processors
- vii. Discount Tickets
- viii. Food Lines
- ix. Products and their pricing

## 2. Inquiries

- i. Event Log (Binnacle)
  - ii. Consultation of Orders
  - iii. Product Inquiry
  - iv. Query of errors log
- The scope of the second module, which is the client for customers will be as follows:
    1. Browse the different products that the restaurant offers.
    2. Generate order of orders.
    3. Interface with credit or debit cards and electronic checks at the time of payment.
  - The scope of the web service will be as follows:
    1. Methods so that the client can generate transactions either by means of cards or electronic checks.

## 4. Functionalities of the administrative module

- When run, the administrative module presents a window where the administrator user must enter valid credentials before entering the application.

### 4.1 Login Screen

- For this screen, the user will enter their login and password to enter the system. Obviously, the user cannot go past this point without credentials [VIEW IMAGES 01 – login and 02 - login on hover].

### 4.2 Home Screen

- The Main Screen that will appear immediately after the user has successfully logged in [VIEW IMAGES 03 – home and 04 - home b].

#### 4.2.1 Options in the Navigation Bar

- Home
- Language
  1. English
  2. Spanish
  3. Italian
  4. German

- Security
  1. Users
  2. Change Password
- Administration
  1. Consecutives
  2. Types of Prices
  3. Credit or Debit Cards
  4. Payment Methods
  5. Discount Tickets
  6. Food Lines
  7. Products
- Inquiries
  1. Orders
  2. Products
  3. Binnacle (Event log)
  4. Error Log
- Log Out Option
- It must display the name of the user currently logged in.
- [VIEW IMAGES 04 - Menu1, 05 - Menu2, 06 - Menu3 and 07 - Menu4].

## 4.3 Security

### 4.3.1 Roles

- There are 4 administrative roles:
  1. Administrator: will have access to all the functionalities of the system.
  2. Security: will only have access to the parts for creating new users, seeing existing users and editing them.
  3. Maintenance: will have access to creating, modifying and deleting records of
    - i. Food Lines
    - ii. Products
    - iii. Discount Tickets
    - iv. Price Types
  4. Queries: will have access only to the queries functionalities that the system has.
- An administrator can only have that role, because it gives to him access to all the functionalities in the app. Because of that, it doesn't make sense to assign to an administrator another role.
- Non administrator users can have a combination of the other existing roles.
- All the users in this client will have access to logging in, changing the language and changing their respective passwords.
- Said roles must be stored manually in the database. There will be no maintenance screen for such roles in the administrative client.

### 4.3.2 Users

- Clicking on this option will bring up the following screen. [VIEW IMAGE 08 - Users].

#### 4.3.2.1 Create User

- Pressing on this option will take the user to the screen for creating new users. [VIEW IMAGE 09 - CreateUsers].

#### 4.3.2.2 View Users

- By pressing on this option, the user will be able to see the list of users. This list will have the option to edit each user. Aspects of a user that can be edited are:
  1. status of a user
  2. role(s) assignment.
- . [VIEW IMAGES 10 . viewUsers 1 and 11 . viewUsers 2].
- Features of this screen:
  1. The update status dropdown menu has the options Active and Inactive.
  2. The roles must be loaded in a checkbox list and mark the ones that the user has assigned.
  3. The update roles button saves the roles assigned to that selected user.
  4. Being an administrator disables the possibility to select more roles for the user.

### 4.3.3 Change Password

- This screen will give the user the possibility to change its current password. The following figure.
- [VIEW IMAGE 12 - ChangePassword].

## 4.4 Administration

### 4.4.1 Consecutives

- The function of this entity is to generate codes that the entities mentioned in the list below need. If a user tries to create a record of one of these entities from the list without first having created a consecutive for that type of element, an error will be generated because those entities need the code that consecutive provides them in order to be created.
- As will be seen at the time the screens of each item in the list below are detailed, they all have in common that they receive a code from consecutives. This code is a field that the user is not going to write when creating one of the items in the list below.
- To explain it with an example, if the first consecutive code for products has not been entered, if you go to the products screen and try to create a product, there will be an error. This is because the table products has as a mandatory field the code that it takes from consecutives which again, the user is not going to write when filling up the form for creating a product. Likewise, if the user has already entered the consecutive 1 for products and the user created the product with code 1, the next time the user creates a product, the code will be generated providing the value +1 (2 in this example) and

the prefix assigned to the first code introduced (if it exists, prefixes are optional). The generation of the next code will be automatically done by the app. When the users select create a product for a second time in this example after creating the first one, the field for its code will automatically say 2.

- This is the list of elements that require a consecutive code:
  1. Types of Prices
  2. List of Credit or Debit Cards
  3. Payment Processors
  4. Food Lines
  5. Products
- When the user navigates to the Consecutives screen, a list with the description and the current consecutive codes for the elements in the list mentioned before will be displayed [VIEW IMAGE 13 - Consecutives].
- Each element of the list will have an Edit button next to it. By clicking on this option, the user will have the ability to edit the information of the code that is going to be used when the next element of this type is created [VIEW IMAGE 14 - ConsecutivesEdit]. The user could edit the following:
  1. Description (the name displayed)
  2. Consecutive (the integer part of the code)
  3. It has a prefix (Boolean that controls if the code has a prefix or not)
  4. Prefix (in this field the user can add a string value for the code, something like "PR-").
  5. If there is a prefix, the prefix part of the code goes before the numerical part.
- At the bottom of the screen, there will be a button for creating a new consecutive. This button will take the user to a screen like this one [VIEW IMAGE 15 - ConsecutivesCreate].
- Validations:
  1. The field to write the Prefix part of the code must be disabled by default. Only if the user chooses the option "It has a prefix" it gets enabled.
  2. If a consecutive code already exists for an entity, and the user tries to use it again, the app must indicate to the user that he can no longer choose this code to be used because it already exists.
  3. A code is defined by its prefix (if it has one) and the integer. That means that once the code 1 for products has been used, it cannot be used again, but the code PR-1 is different from the code 1 and can be used.
  4. It is not necessary to validate that the integer part of the code selected by the user follows a lineal increment of +1 and starts at 1. This means that for example, a user could create as the first consecutive code for products code 99, and from there the automatic assignment of the following code for products will go from there adding +1 and using the same prefix. Later on, the user could edit the consecutive code that is going to be used in the creation of the next product and write code 1 + the prefix that was being used before. There is no problem as long as there are no duplicates. If by doing this the auto assignment of the code for products arrives to a duplicate value as the next

code to be used, the app must inform that the user has to edit this field before creating the new product.

#### **4.4.2 Types of Prices**

- From this option you can create the different types of prices that can be assigned to products. For example, assigning the types of regular, normal or enlarged to a combo.
- On this screen the user will see a list of the existing Types of prices [VIEW IMAGE 16 - TypesOfPrices]. Each item in the list will have the option to:
  1. Edit: Modify the information of the element selected in the list. The code is not modifiable [VIEW IMAGE 17 - TypesOfPricesEdit].
  2. Delete: A confirmation is requested that you want to delete the selected item from the list, and you proceed to receive a confirmation [VIEW IMAGE 18 - TypesOfPricesDelete].
- At the bottom of the screen there will be a button to create a new item [VIEW IMAGE 19 - TypesOfPricesCreate].
- Validations:
  1. The code must be self-generated and not modifiable.

#### **4.4.3 List of Credit or Debit Cards**

- From this option, new credit or debit cards can be created, so that later they can be assigned to the corresponding processor that handles the transactions.
- On this screen the user will see a list of the existing Credit or Debit Cards [VIEW IMAGE 20 - CreditOrDebitCards]. Each item in the list will have the option to:
  1. Edit: Modify the information of the element selected in the list. The code is not modifiable [VIEW IMAGE 21 - CreditOrDebitCardsEdit].
  2. Delete: A confirmation is requested that you want to delete the selected item from the list, and you proceed to receive a confirmation [VIEW IMAGE 22 - CreditOrDebitCardsDelete].
- At the bottom of the screen there will be a button to create a new item [VIEW IMAGE 23 - CreditOrDebitCardsCreate].
- Validations:
  1. The code must be self-generated and not modifiable.

#### **4.4.4 Payment Methods (Payment Processors)**

- This will work as the payment methods that can be used by customers to pay for their orders. These will be in charge of processing the all the possible ways of paying for an order, electronic cards, checks and cash. Therefore, there may be different processors that process credit cards or checks, but there can only be one payment option that says cash.
- On this screen the user will see a list of the existing Payment Methods [VIEW IMAGE 24 - PaymentMethods]. Each item in the list will have the option to:
  1. Edit: Modify the information of the element selected in the list. The code is not modifiable [VIEW IMAGE 25 - PaymentMethodsEdit].

2. Delete: A confirmation is requested that you want to delete the selected item from the list, and you proceed to receive a confirmation [VIEW IMAGE 26 - PaymentMethodsDelete].
- At the bottom of the screen there will be a button to create a new item [VIEW IMAGE 27 - PaymentMethodsCreate].
  - Validations:
    1. The code must be self-generated and not modifiable.
    2. When creating a new payment method, if it is chosen that said registration is that of the type of credit or debit cards, the option to assign new credit cards must be enabled. If it is not, it must be disabled.
    3. There should only be one means of payment that is cash. If another type of the same type is entered, the app must indicate to the user that this type of payment already exists.
    4. There can only be one active processor for credit or debit cards and one for electronic checks. If the app is using a processor for credit cards and the user wants to change it, the user first needs to edit it and deactivate it. After that, he can choose to activate another one. If a client tries to buy something while there is no active processor for the payment method that the user elected, the app must show an error.

#### **4.4.5 Discount tickets**

- The Discount Tickets screen allows the user to create coupons that can use when paying for an order to get a discount. The discount can be a percentage or a monetary amount.
- On this screen the user will see a list of the existing Discount Tickets [VIEW IMAGE 28 - DiscountTickets]. Each item in the list will have the option to:
  1. Edit: Modify the information of the element selected in the list. The code is not modifiable [VIEW IMAGE 29 - DiscountTicketsEdit].
  2. Delete: A confirmation is requested that you want to delete the selected item from the list, and you proceed to receive a confirmation [VIEW IMAGE 30 - DiscountTicketsDelete].
- At the bottom of the screen there will be a button to create a new item [VIEW IMAGE 31 - DiscountTicketsCreate].
- Validations:
  1. The code must be self-generated and not modifiable.
  2. An integer value must be entered in the available quantity field.

#### **4.4.6 Food Lines**

- It will manage the different types of food that the store can offer, for example, Combos, Desserts, Salads, among others.
- On this screen the user will see a list of the existing Food Lines [VIEW IMAGE 32 - FoodLines]. Each item in the list will have the option to:

1. Edit: Modify the information of the element selected in the list. The code is not modifiable [VIEW IMAGE 33 - FoodLinesEdit].
  2. Delete: A confirmation is requested that you want to delete the selected item from the list, and you proceed to receive a confirmation [VIEW IMAGE 34 - FoodLinesDelete].
- At the bottom of the screen there will be a button to create a new item [VIEW IMAGE 35 - FoodLinesCreate].
  - Validations:
    1. The code must be self-generated and not modifiable.

#### **4.4.7 Products**

- This screen corresponds to the maintenance of the different products that the restaurant can offer, be it combos, soft drinks, hamburgers, desserts, among others. These products will be those that the user can purchase when entering the E-Food client for customers.
- On this screen the user will see a list of the existing Products [VIEW IMAGE 36 - Products]. Each item in the list will have the option to:
  1. Edit: Modify the information of the element selected in the list. The code is not modifiable [VIEW IMAGE 37 - ProductsEdit].
  2. Add Price: By clicking on this option, you can add the prices that the product may have, and this screen will appear to carry out this action [VIEW IMAGES 38 – ProductsAddPrice, 39 – ProductsAddPriceEdit and 40 - ProductsAddPriceDelete]. Within this window you will have the option to view the prices of the products, add new ones, modify and delete the existing prices. The link back to Products returns to the screen where the list of products is.
  3. Delete: A confirmation is requested that you want to delete the selected item from the list, and you proceed to receive a confirmation [VIEW IMAGE 41 - ProductsDelete].
- At the bottom of the screen there will be a button to create a new item [VIEW IMAGE 42 - ProductsCreate].
- Validations:
  1. The code must be self-generated and not modifiable.

#### **4.5 Log**

- It will be an internal process which is invoked every time a user inserts, removes or modifies a record and deletes an order.
- The information that needs to be registered is:
  1. the user
  2. date
  3. time



4. the registration code
5. the description of the action that was performed. For example: insertion of a new product, modification of a food line.

## 4.6 Queries

### 4.6.1 Event Log (Binnacle)

- In this screen it can be consulted for:
  1. a user and
  2. a date range

the actions that that user has made on the system. The query lists:

1. the event date
  2. event description
  3. registration code.
- [VIEW IMAGE 43 - Binnacle].

### 4.6.2 Orders

- It consists of consulting by:
  1. date range and
  2. by state the orders that are in the system

The query returns:

1. the order number
2. date
3. amount

There are the following states:

1. Processed
  2. Canceled
  3. In Progress
- When querying for In Progress status, the option to delete elements will be enabled so that the user can delete that orders from the database corresponding to the checkboxes selected in the list of in progress orders. For this, the user will select the orders and delete the ones that are selected. This will be done using the checkboxes in a DataGrid [VIEW IMAGE 44 - Orders].

#### 4.6.3 Products

- It consists of selecting a food line and listing all the products that are under that line. This list will present:
  1. the product code
  2. name
  3. prices that the product may have
- [VIEW IMAGE 45 - Products]

#### 4.6.4 Errors

- It consists of displaying a list of the errors that have been registered in the application (errors in the database, in the backend, in the administrative client and in the customers client). The query will have the option to filter by date range. The list will display:
  1. date
  2. time
  3. error number
  4. error message
- [VIEW IMAGE 46 - errors]

### 5. Functionalities of the Client for Customers

- This client is intended to be run separately from the administrative client and will be used by the customers of the restaurant. When the customer enters the application, they will be presented with a home screen like this [VIEW IMAGES 47 - ClientsHome1 and 48 - ClientsHome2].
- The mockups don't show it, but the customers client must be able to change the language too.

#### 5.1 Product Search

- There are two ways in which a customer can search for products:
  1. The customer selects a food line (the existing options will be inside a dropdown box) and will press the search button. When such a search occurs, the app redirects the customer to a new page that presents the list of the products under the category that the customer selected [VIEW IMAGE 49 - ClientsProductSearch1]. The list will present:
    - i. the product code
    - ii. name
    - iii. price
    - iv. image (optional)
    - v. an option to see the details of an element in the list

If the customer clicks on see details, all the information of that product will be presented in a new view. The screen should be very similar to the following [VIEW IMAGE 50 - ClientsProductSearch2]. The add button inserts said product into the customer's shopping cart. The app must validate that the customer entered a quantity and a price.

2. The customer selects a food line and enters a value in the text box next to it and presses the search button. If this search returned results, a screen like the following [VIEW IMAGE 50 - ClientsProductSearch2] should be presented to the customer. On this screen the customer will be able to see the product in detail and will have the option of adding that product to the cart [VIEW IMAGE 50 - ClientsProductSearch2]. If the search did not return anything, it should indicate to the customer a message that the search did not return anything [VIEW IMAGE 51 - ClientsProductSearchNoResult].

## 5.2 View cart

- By clicking on this option the customer will be able to see the content of their order. The customer will have the option to modify the quantity for a product and also will be able to delete that product. The screen must present what the cart has in the form of a list and two options for each element: delete and modify [VIEW IMAGES 53- ClientsViewCart editQuantity and 54 - ClientsViewCart deleteProduct].

## 5.3 Payment

- It will indicate that the customer wants to finish the order and make the respective payment. The payment will consist of three steps:
  1. Collection of customer data [VIEW IMAGE 55 - ClientsViewCart pay1]:
    - i. the name
    - ii. surname
    - iii. telephone number
    - iv. shipping address
    - v. an option to enter a discount ticket. If a ticket code is introduced, the app will validate if the promotion number exists and if there is any discount that applies.
  2. The customer will have the option to use Facebook or Twitter to authenticate: The customer can choose, if he does not want to create an account to pay, he can use a login with Facebook or Twitter, for this he can use his personal accounts and log in, which will allow him to carry out the process without the need of having to create an account.
  3. Collection of payment data: in this step the customer will indicate the form of payment that he will use to pay for the order. It is required to collect the payment method that will be used, which will be one of the ones on the list of the payment methods that are in the database. This screen will also present to the customer if the discount ticket that he entered was correct and if the discount was applied to him. It is important that the customer knows this before processing [VIEW IMAGE 56 - ClientsViewCart pay2]. If the payment

option does not require verification against a processing entity (CASH), the client is redirected to a final confirmation screen [VIEW IMAGE 60 - ClientsViewCart pay6]. If the option requires verification, it will be redirected to the corresponding page and this will depend on the following.

4. If the method is credit or debit card, the corresponding screen will be presented [VIEW IMAGE 57 - ClientsViewCart pay3]. This screen captures:
  - i. the card number
  - ii. card expiration date
  - iii. card type (Visa, MasterCard, etc.) which will be the types associated with the processor and CVV.
  - iv. CVV
  - v. The amount for which the order will be processed

When the customer enters all the fields, they can press the process button. When this action is carried out, the card must be verified against the payment method that is currently active that is of the credit and debit card type and call the method assigned in the method field. If the processing was successful, the customer is directed to the final screen where a final confirmation is presented to the customer [VIEW IMAGE 60 - ClientsViewCart pay6]. Otherwise, the customer will be presented with a screen indicating that the transaction failed [VIEW IMAGE 58 - ClientsViewCart pay4].

5. If the method is electronic checks, the following screen will appear [VIEW IMAGE 59 - ClientsViewCart pay5]. It captures the check number and the checking account. The amount for which the order will be processed will also be captured. When the required fields are entered and the process button is pressed, it will be verified against the payment method that is active and that is of the electronic check type. If the processing was successful, the customer is directed to the final screen where a final confirmation is presented to the customer [VIEW IMAGE 60 - ClientsViewCart pay6]. Otherwise, the customer will be presented with a screen indicating that the transaction failed [VIEW IMAGE 58 - ClientsViewCart pay4].
6. Final Confirmation: final step where the information stored in the previous steps and two options are presented in detail to the customer. One to confirm the order as completed and the other to cancel the order [VIEW IMAGE 60 - ClientsViewCart pay6].
  - i. When the customer clicks on Save Order, the customer will be presented with a message that the order was saved successfully and the customer will be presented with the order number.
  - ii. When the customer clicks on Cancel Order the customer will be shown a message that says that the order was canceled.
  - iii. Both confirmations will have an option for the customer to return to the home screen.

## 6 Web Service

- It is an application that will serve as the payment processor when the payment method is Credit Card or Electronic Check. For the functionality of said Web service, two methods are required:

### 6.1 Credit or Debit Card

- It consists of processing the credit or debit card information that the customer entered when processing the order. For this, the WEB service needs to receive the following parameters.

| Parameter       | Datatype | Description            | Example     |
|-----------------|----------|------------------------|-------------|
| cardNumber      | Integer  | Card number to process | 1.23457E+15 |
| expirationMonth | Integer  | Expiration month       | 6           |
| expirationYear  | Integer  | Expiration year        | 2007        |
| CVV             | Integer  | Card verification code | 8542        |
| cardType        | String   | Card type              | Visa        |
| total           | Decimal  | Monto del pedido       | 4500        |

- When the necessary parameters are collected, it must be verified that all the previous information exists in the database of the web service. If it's a debit card, it must be confirmed that the customer has the necessary funds to make the transaction. To verify this in the web service database, if the cardType is equal to Debit, then it is verified that there are enough funds to pay for the transaction. If there are enough funds, the total amount of the order is discounted from the balance that the customer owns.
- After making these verifications, the status of the transaction, whether it failed or not, must be returned outside the method. For this, a string will be returned and can be one of the following:

| Return   |
|--|
| 0; Successful Transaction                      |
| -1; Invalid card number                        |
| -2; Expiration date or invalid or expired card |
| -3; CVV incorrect                              |
| -4; Card type not supported                    |
| -5; Insufficient funds                         |

### 6.2 Electronic Check

- It consists of processing the check information that the customer entered when processing the order. For this, the WEB service needs to receive the following parameters.

| Parameter   | Datatype | Description             | Example     |
|-------------|----------|-------------------------|-------------|
| checkNumber | String   | Check number to process | A-20507     |
| account     | Integer  | Check account number    | 8.00016E+13 |
| total       | Decimal  | Order amount            | 4500        |

- When the necessary parameters are collected, it must be verified that all the previous information exists in the database. It must be verified that the client has sufficient funds in the account of that check and if it does, the amount of the order is discounted to the balance that the client owns.
- After making these verifications, the status of the transaction, whether it failed or not, must be returned outside the method. For this, a string will be returned and can be one of the following:

| Return                     |
|----------------------------|
| 0; Transaction successful  |
| -1; Invalid check number   |
| -2; Invalid account number |
| -3; insufficient funds     |

## 7 Change of Language

- This function will allow all users to change the language of the program in its entirety, for this they must investigate how to do it and that the language can be changed in all sections of it, the languages are English, German, Italian and Spanish.

## 8 Design

- In the administrative client the login screen is going to be called Login.aspx.
- In the administrative client the home screen is going to be called Default.aspx.
- The app needs to use Bootstrap to generate the entire design. (REACTIVE)
- The app must use masterpages to create the template that will be used to create the other pages. This in order to maintain design consistency on all pages.
- the app must use validation controls for fields that require it.
- The app must use sorting and pagination when presenting lists. (DATATABLES)

## 9 Error Page

- The app must present the user with an error page every time one of them appears and record the error in the database. Registration is required, the error number, date and time and the corresponding message [VIEW IMAGES 61 - AdminClient ERROR and 62 - UserClient ERROR].

## **10 Database**

- The database must have stored procedures that execute the application logic. To do this, it is necessary to build procedures that insert, modify, update, and query the database.

## **11 WEB Service Database**

- It is necessary to build a database to handle the credit/debit cards and checks. This is a separate database and is unrelated to the project database.

## **12 Classes or Layers**

- The application must have the following:
  1. A class project where the database access logic (DAL) resides.
  2. A class project where the application's business logic (BLL) resides.
- Classes: it is necessary to work with a double encryption in MVC, one in JavaScript and the other in Visual Studio or any other programming language, in such a way that the JavaScript code is not seen in any of the pages.

## **13 Cart Number**

- It is necessary to implement a logic to assign the cart number to the order that a customer is placing and not lose that number during the execution of the customer's visit to the site.