**CONSIGNMENT STORE REPORT**

### **PREPARED FOR INSTRUCTOR**

Michael Hrybyk

### **PREPARED BY STUDENTS**

Vu Duy Hau Nguyen (Project Lead)

Vasu Joshi

Apr 10, 2019

# **1. Project Description**

**1.1. Project Overview**

This project is carried out for developing a general management system which can be easily integrated and further optimized to fit the need of any consignment stores. The system keeps track of all consignments information (owners, responsible employees, status ...), as well as generate the consignment results at the end of each consignment period. Moreover, it also provides useful insights of the current consignment business (e.g. current usable budget of the store or total money which the store temporarily keeps for consignors). Therefore, the admin or store owner can make important decisions at the right time.

The ideal flow of the system is introduced next. Consignors bring their items to a consignment store, price and consign them with the help of store employees. Then, the store tries to sell them in one consignment period (e.g. 2 months). At the end of consignment period, the system will generate consignment results for all consignors of that specific period. A consignment fee will be applied (e.g. 30% price of sold items). Finally, consignors can go to the store to get back their consignment results (can be money and/or unsold items). In case, consigned items were lost in the store, the system will record them for later analysis and treats them as sold items (return money back to consignors).

**1.2. Deliverables**

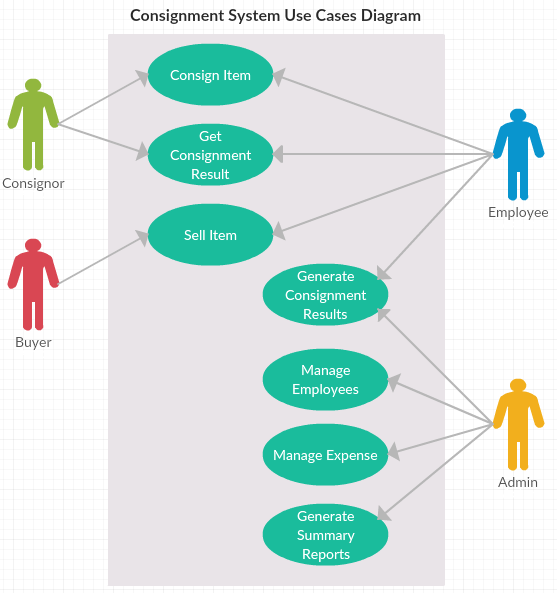
The general management system in this project will be implemented in a Windows Forms application which is written in C# language and uses Microsoft SQL Server as a DBMS of the application.

**1.3. Scopes**

This project is mainly focus on manage the main flow of the consignment cycle, from received consigned items to return the consignment results. Because of the time and availability of equipment constraints, this project has some limitations as shown below:

* Customer (buyer) information is not considered in this project.
* Employees management such as timesheets, payroll is also skipped. It is simplified as one big expense in **Salary** category every month.
* The consigned items are kept simple. It can have more details and be categorized in real world system (e.g. item in ***book*** category may have more attributes such as **title**, **ISBN**, **year** and **author**).
* No tax systems are applied to the system.
* Identification and Point-Of-Sale system which usually can be integrated to the system such as barcode labeling (for identification of the consigned items), barcode reader and receipt printer (for selling process) are omitted. In this project, the system accepts only input data from mouse and keyboard.

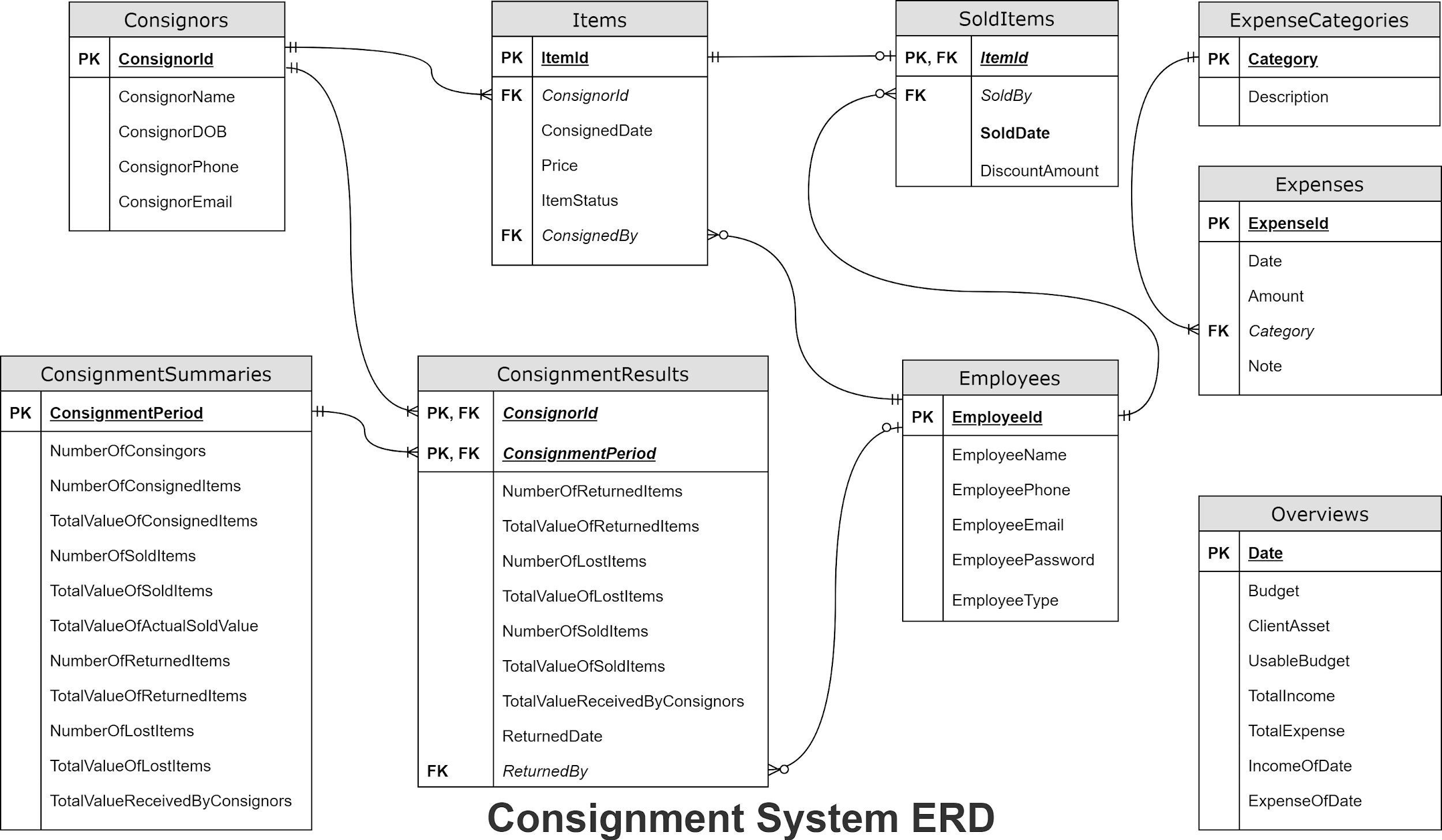
# **2. Process Model**



|  |  |  |
| --- | --- | --- |
| **Use Case** | **Actor** | **Description** |
| **Consign Item** | *Consignor, Employee* | A consignor brings item to store to consign with the help of an employee |
| **Sell Item** | *Employee, Buyer* | An employee sells consigned item to a buyer |
| **Generate Consignment Results** | *Admin, Employee* | Admin generate the consignment results of a specific period with the help of employees (e.g. find all unsold items) |
| **Get Consignment Result** | *Consignor, Employee* | At the end of consignment period, a consignor gets the consignment result with the help of an employee |
| **Manage Employees** | *Admin* | Manage the employees of the store (e.g. grant system access for an employee) |
| **Manage Expense** | *Admin* | Can make an expense (e.g. salary of employees, utility bills) |
| **Generate Summary Reports** | *Admin* | Generate various reports (e.g. daily income, consignment period summary) |

# 

# **3. Database Model**



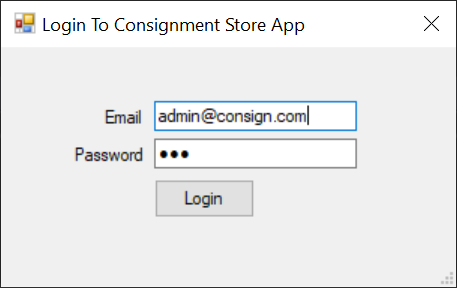
Tables Overview:

* **Consignors:** stores information of consignors.
* **Employees:** stores information of employees and their **types** which can be one of three states:“**employee**” and “**admin**”.
* **Items:** stores information of each consigned item (e.g. owner, employee who processes the consignment ...). Note that, item can be at one of four **states** (“**unsold**”, “**sold**”, “**lost**” and “**returned**”).
* **SoldItems:** stores information of each item sold and its **DiscountAmount** if applicable
* **ConsignmentSummaries:** stores overview data of each consignment period:
  + “**NumberOfConsignors**”: total number of consignors.
  + “**NumberOfConsignedItems**”: number of consigned items.
  + “**TotalValueOfConsignedItems**”: sum of all consigned items’ prices.
  + “**NumberOfSoldItems**”: number of sold items.
  + “**TotalValueOfSoldItems**”: sum of all sold items’ prices.
  + “**TotalValueOfActualSoldValue**”: “**TotalValueOfSoldItems**” - sum “**DiscountAmount**”.
  + “**NumberOfReturnedItems**”: number of items which will be returned to their consignors.
  + “**TotalValueOfReturnedItems**”: sum of all returned items’ prices.
  + “**NumberOfLostItems**”: number of lost items.
  + “**TotalValueOfLostItems**”: sum of all lost items’ prices.
  + “**TotalValueReceivedByConsignors**”: sum of all money which will be returned to consignors.
* **ConsignmentResults:** stores information of consignment results for each consignor at the end of their consignment period. Note that, at the beginning “**ReturnedDate**” and “**ReturnedBy**” are **null.** They will be filled with data after a consignor receiving the consignment result.
* **ExpenseCategories:** stores some common category of expense (e.g. salary, utility bills).
* **Expenses:** stores details of every expense.
* **Overview:** stores some overview data for each date:
  + “**Budget**”: total money system can access.
  + “**ClientAsset**”: estimated total money of consignors that system keeps.
  + “**UsableBudget**”: equals “**Budget**” - “**ClientAsset**”, total money system can freely use.
  + “**TotalIncome**”: total income from the beginning.
  + “**TotalExpense**”: total expense from the beginning.
  + “**IncomeOfDate**”: total income of that day.
  + “**ExpenseOfDate**”: total expense of that day.

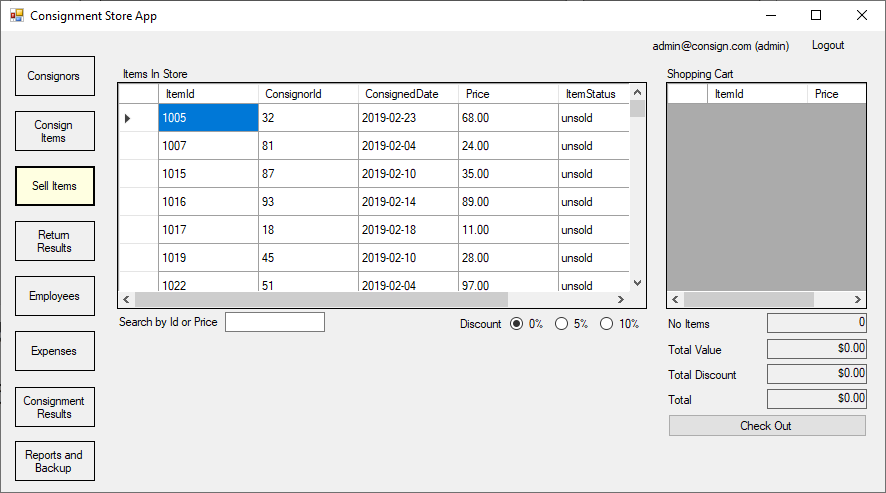
# **4. User Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Screen** | **User Type** | **Description** |
| **1** | **Login** | *Admin, Employee* | Use to sign up the system |
| **2** | **Manage Consignors** | *Admin, Employee* | Use to create, view and update consignor information |
| **3** | **Manage Employees** | *Admin* | Use to create, view and update employee information (not yet implemented) |
| **4** | **Manage Expenses and Expense Categories** | *Admin* | Use to create, view expense and expense category information (not yet implemented add and update expense category) |
| **5** | **Consign Items** | *Admin, Employee* | Use to make consignment with consignor |
| **6** | **Sell Items** | *Admin, Employee* | Use to sell consigned items |
| **7** | **Manage Consignment Results** | *Admin* | Use to generate and publish consignment results and consignment summaries |
| **8** | **Return Consignment Results** | *Admin, Employee* | Use to return consignment results to consignors |
| **9** | **View Reports** | *Admin* | Use to view overview reports of business, backup and restore data (not yet implemented restore from xml files) |

Deal to the complex of business model and time constraint. Some functions that have color red above are not implemented yet.

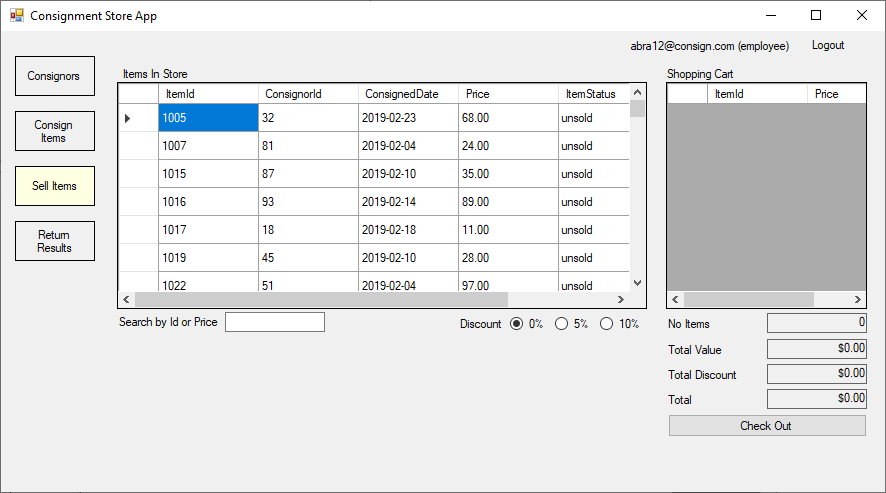


Login

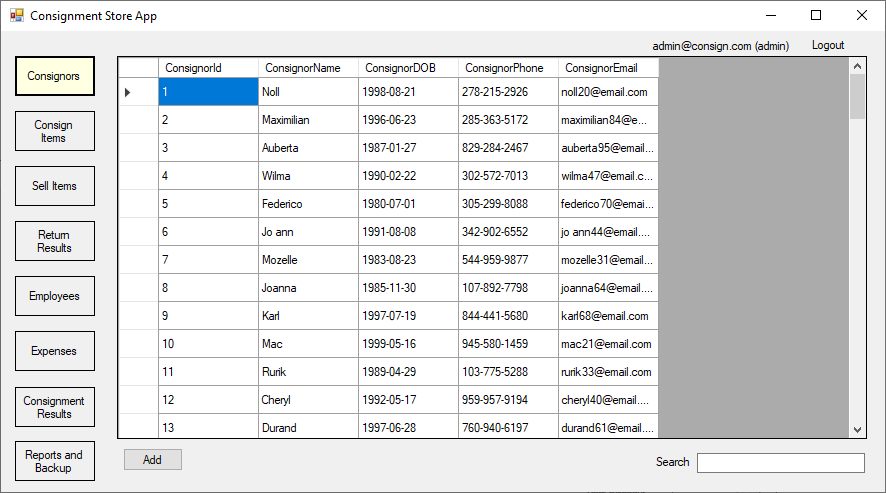


Sell Items - default screen after logged in (Logged in as **admin** can see all functions).

* Show all “**unsold**” items
* Double-click on item in **Items in Store**, it will be move to **Shopping Cart** and cart details will be dynamically displayed below.
* Double-click on item in **Shopping Cart**, it will be move to **Items in Store** and cart details will be dynamically displayed below.
* Can search for items by their Id and Price
* Select Discount radio button (**0%, 5%, 10%**) to apply Discount for each items in **Shopping Cart**
* Click **Check Out** button to finish selling process

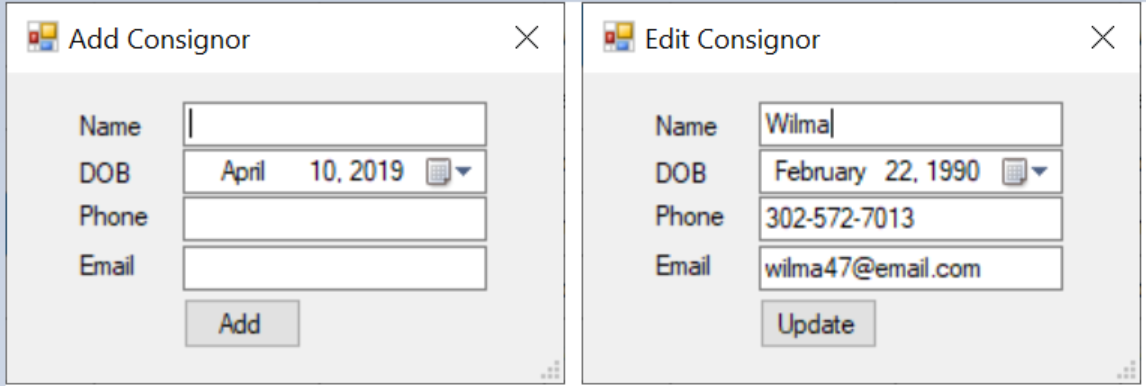


Sell Items - default screen after logged in (Logged in as **employee** can only see 4 functions)

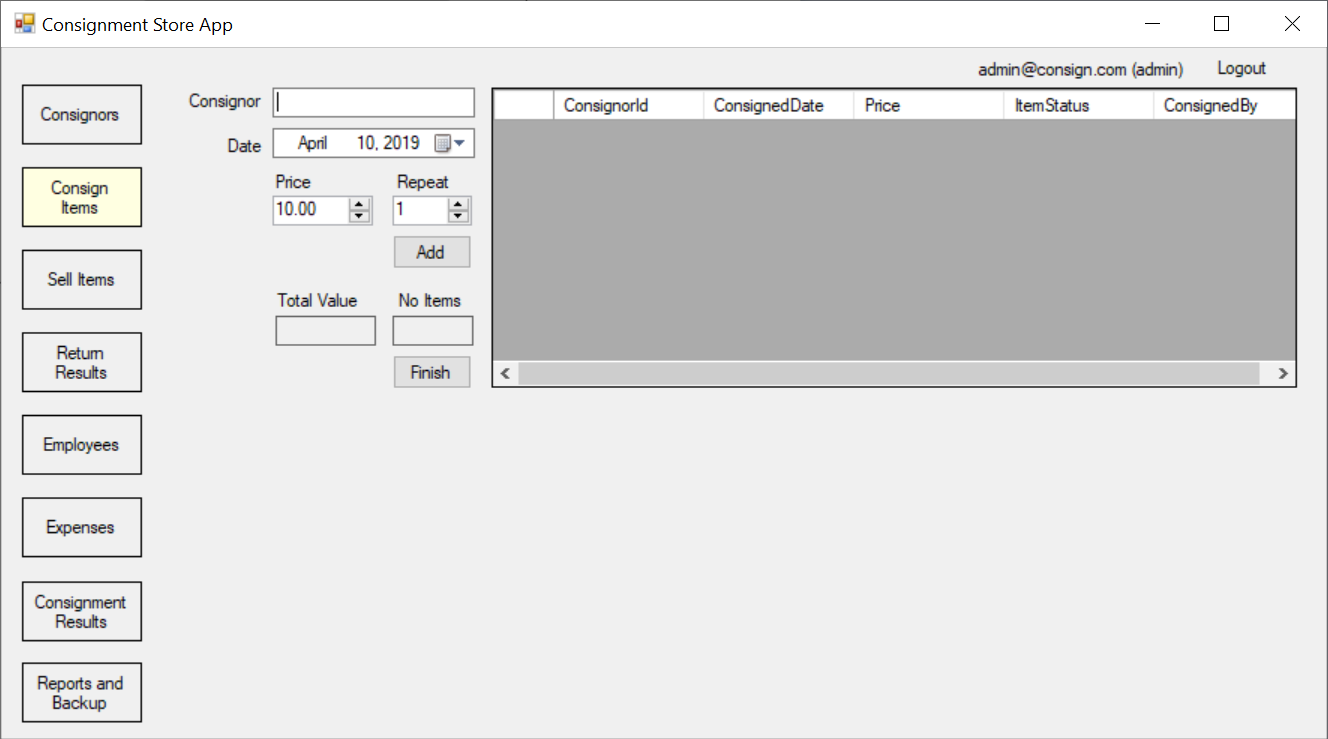


Manage Consignors

* Double-click on a consignor to edit.
* Click Add button to add new consignor
* Can search by Email, Phone, Name and DOB

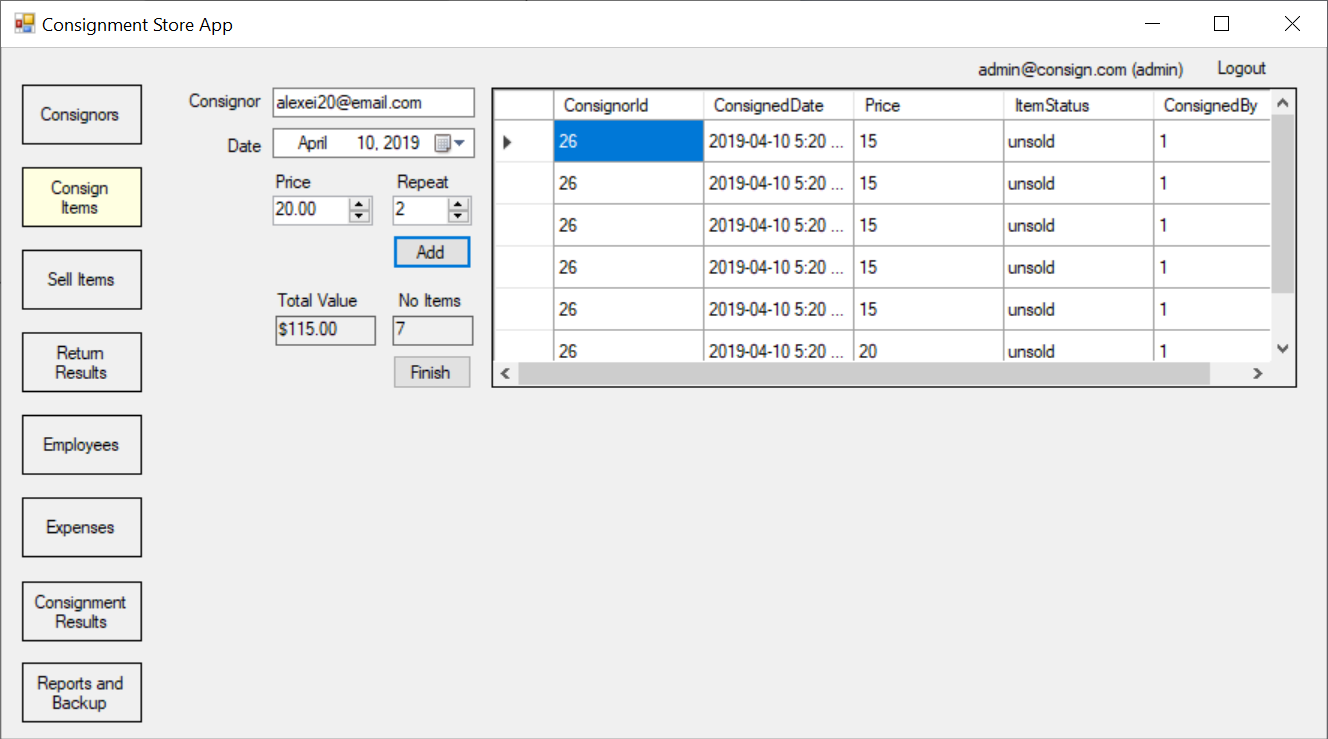


Add and Edit a Consignor Forms

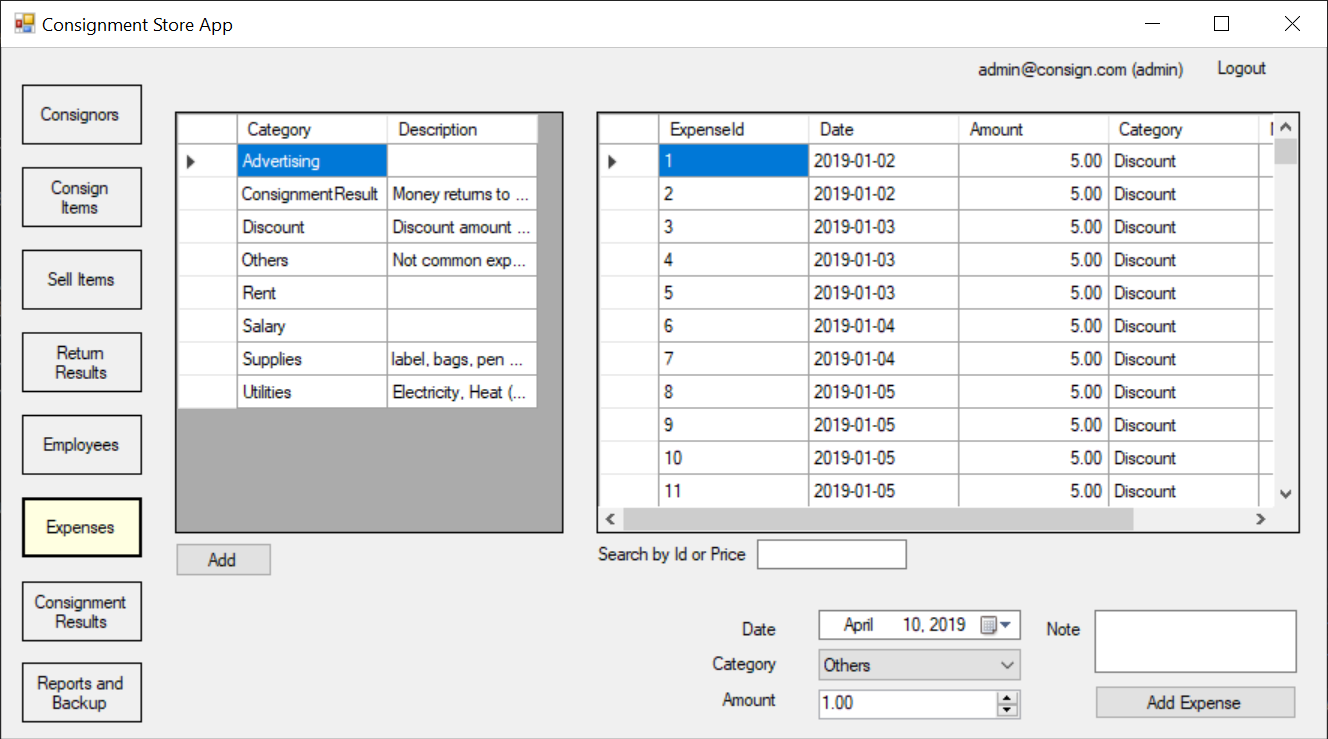


Consign Items

* Date cannot in the past
* Enter email of a consignor who want to consign (consignor textbox will give hints about the consignor email, must choose between available emails)
* Click **Add** button to add consigned items to the left table
* Can set **repeat** number (e.g. consign 10 items with prices $10.00), before click **Add** button
* Click **Finish** button to finish the consignment process

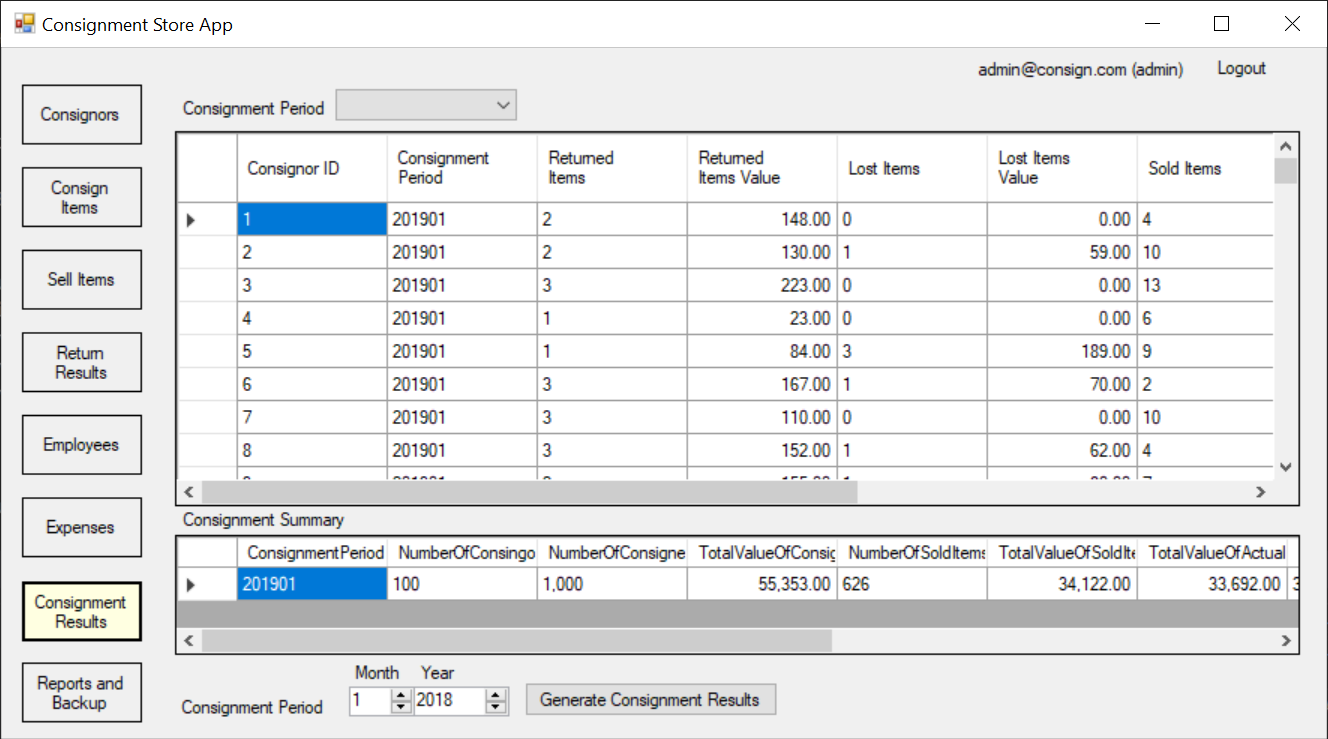


Consign Items with added items (click **Finish** button to save to database)



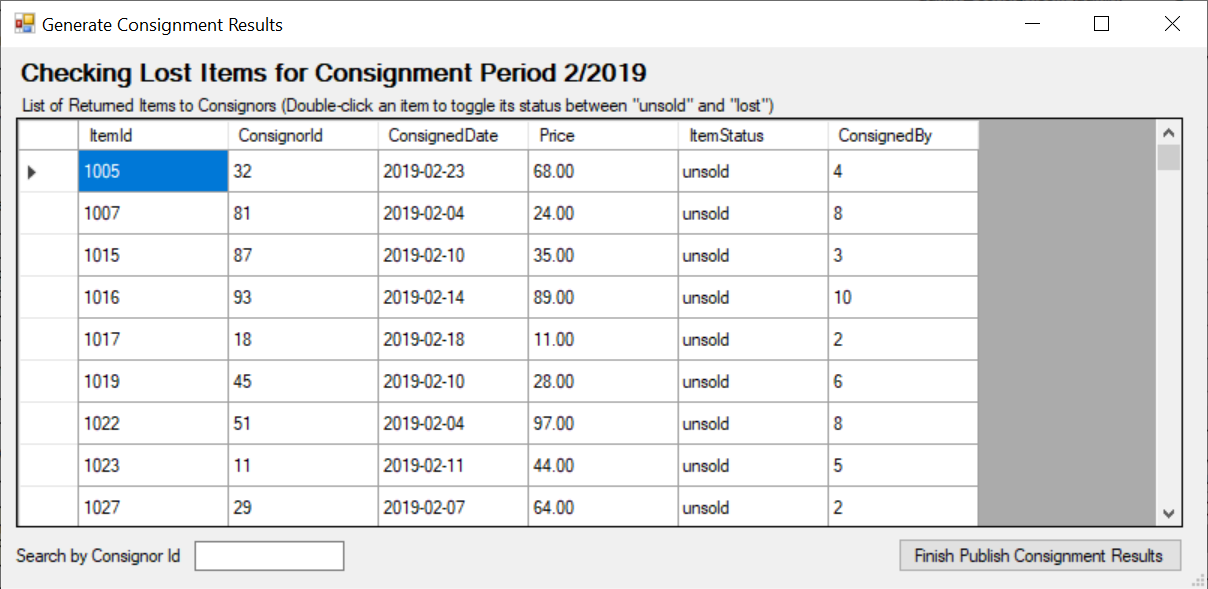
Manage Expenses

* Display all Expense Categories and Expenses
* Date cannot in the past
* Choose Category from the combo box, enter Amount and Note (optional) then click **Add Expense** button to add an expense
* Can search by Category, Date and Note



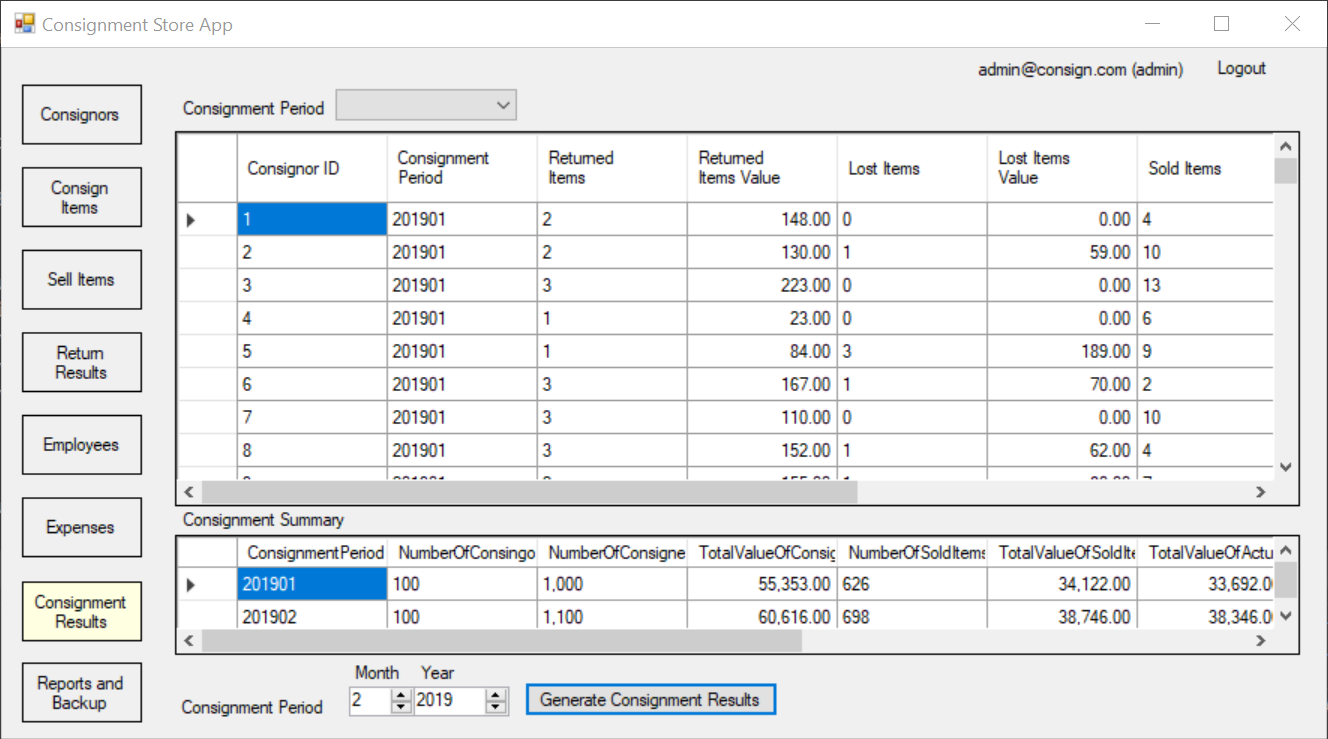
Manage Consignment Results and Consignment Summaries

* Display all Consignment Results and Summaries by default (can choose to view specific consignment period by select in the combo box at top)
* Choose suitable **Month** and **Year**, then click **Generate Consignment Results** button to start the process of generate consignment results and summary (Note that - there is only one consignment summary for Jan 2019)
* Now generate consignment results and summary for Feb 2019

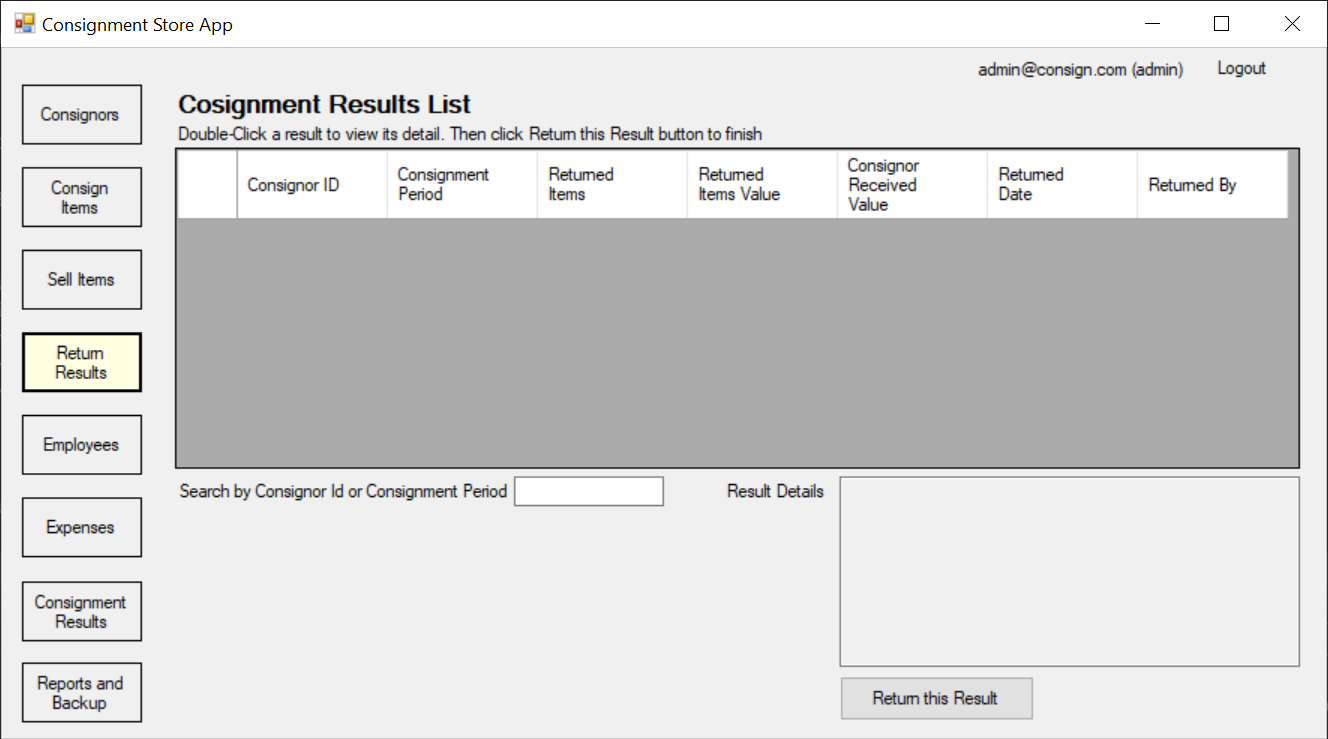


Generate Consignment Results and Summary Form

* Double-click on item if it is lost.
* Click **Finish Publish Consignment Results** to finish the process (the process will take a bit of time to finish) and the result is shown below. Note that, after the process all “unsold” items from Feb 2019 are changed to “returned” or “lost” items, so they do not appear in **Sell Items** anymore.

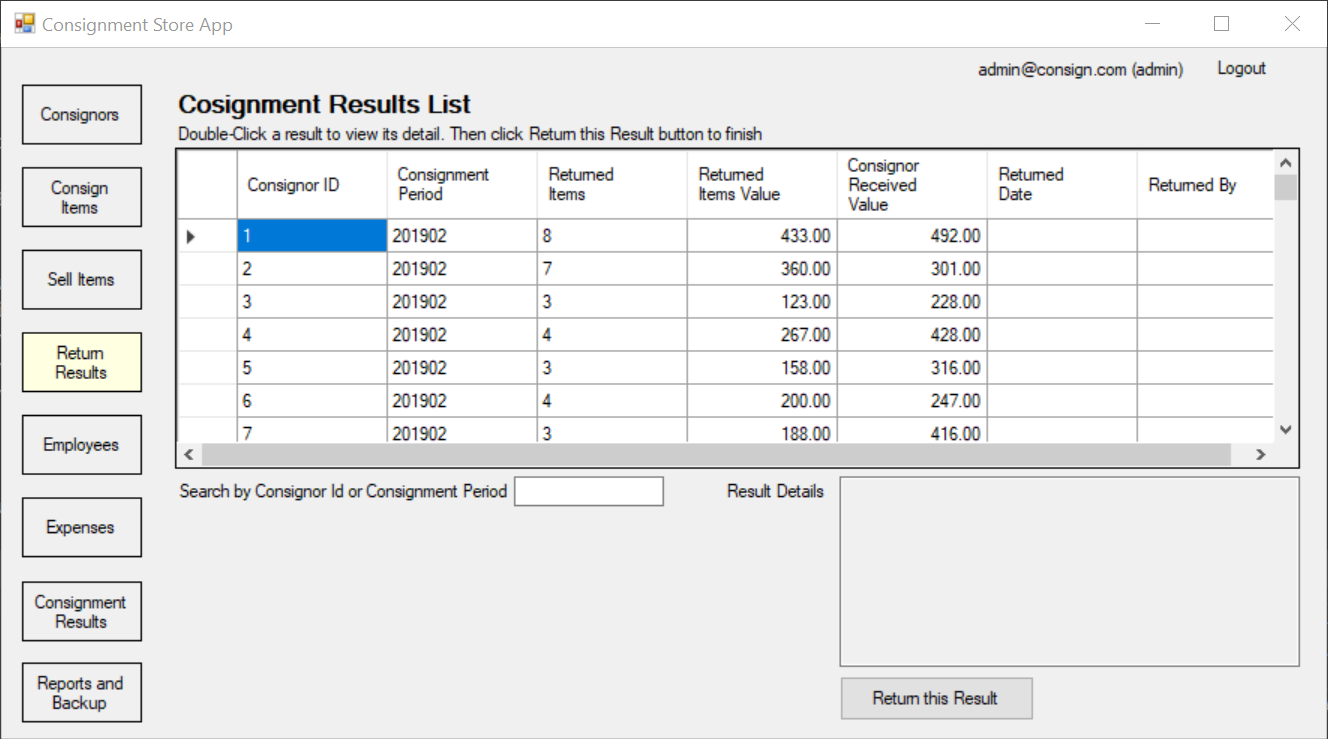


Consignment Results and Summary for Feb 2019 are published

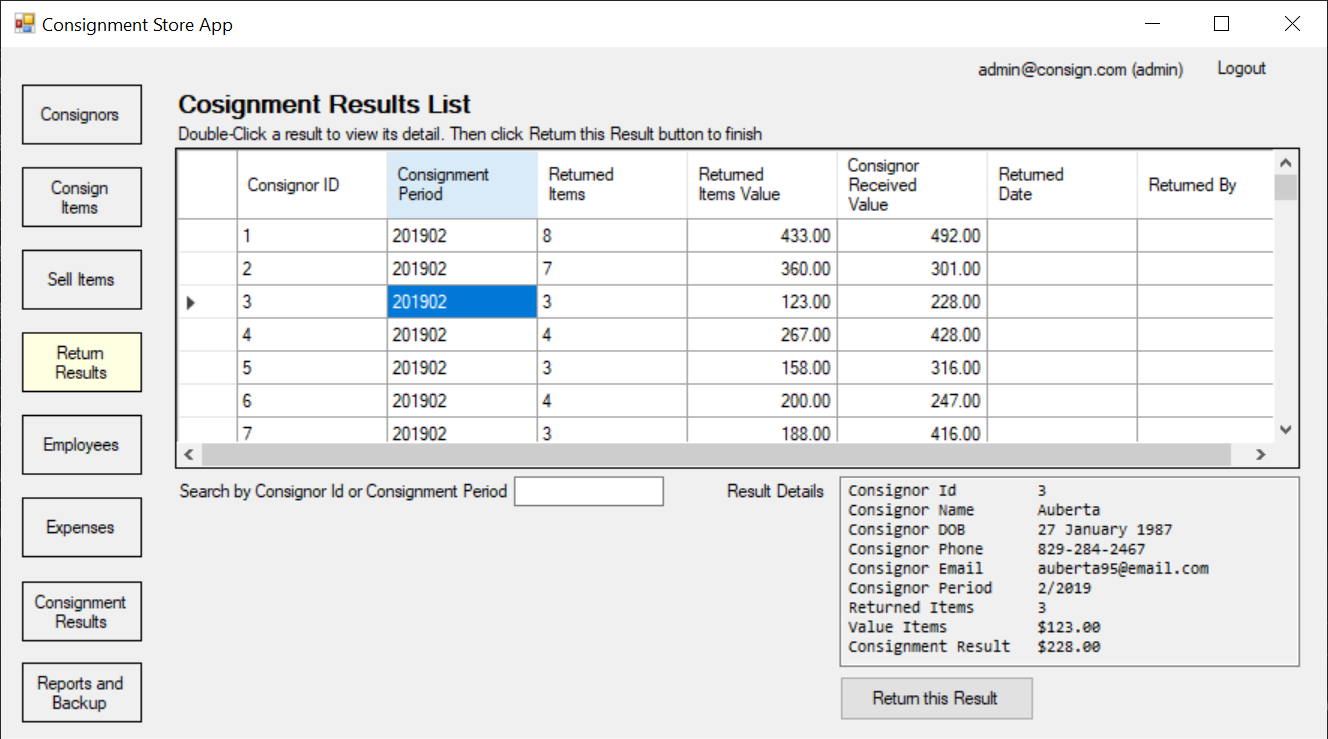


Return Consignment Results

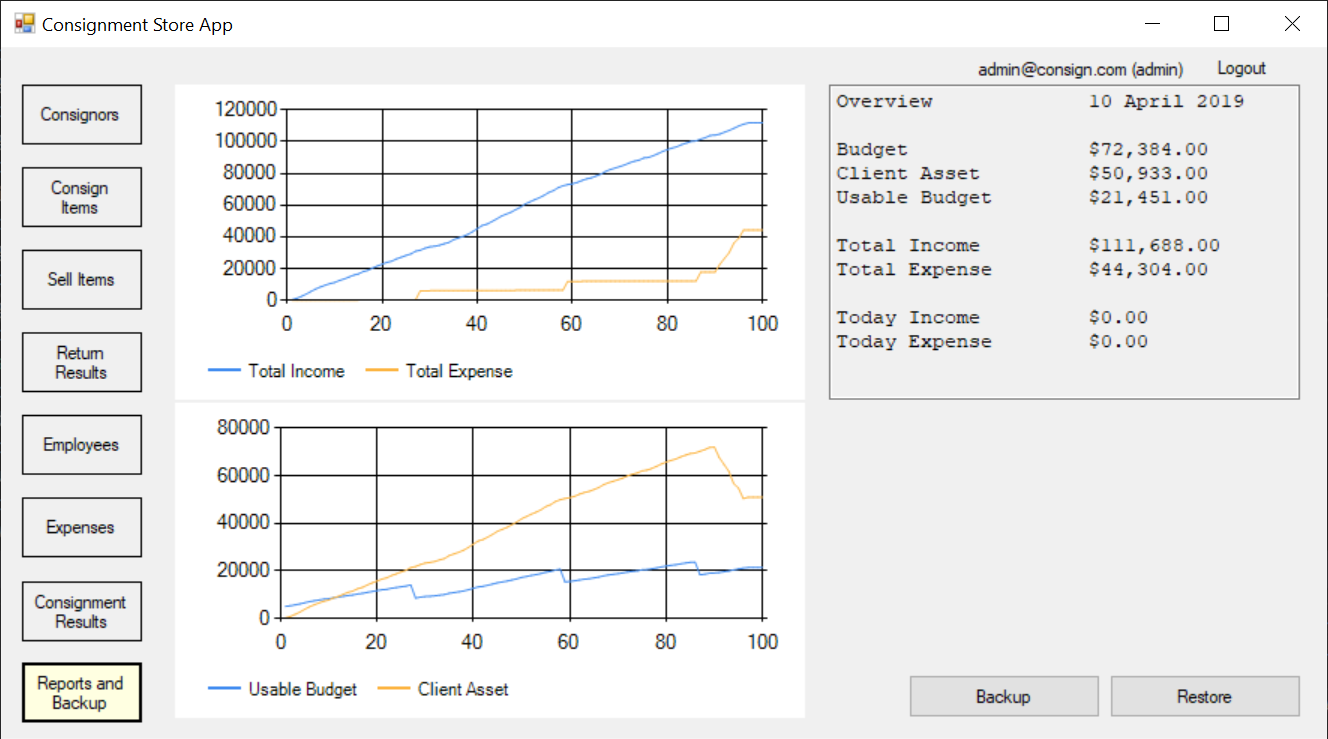
Note that there is no consignment results to returned, because all consignment results of Jan 2019 were returned and there is no published consignment results of Feb 2019 yet. After clicking **Finish Publish Consignment Results** in **Generate Consignment Results and Summary Form** above and enter **Return Consignment Results** again, we will see some consignment results need to return as shown below (if nothing shows up, try to enter some search key to refresh the view)



Return Consignment Results



* Double-click on a result to view its details (consignor info, number of returned items and money the consignor can receive)
* Click **Return this Result** button to finish the process
* Can search by Consignor Id and Consignment Period



Reports and Backup Screen

* Quickly view the business crucial info such as current Budget, Usable Budget, Total Income, Total Expense, Today Income and Today Expense. Note that, when make an expense or a sale, and then return to **Reports and Backup Screen** we can see all the data update.
* The top chart shows the **Total Income** and **Total Expense** overtime. It is matched with the seed data, we can see that there are 3 small increases in expense at the end of each months (seed data generate a lot of expense in date 28th each month such as Salary, Rent, Utilities ...). And the big increase at the end matched with the returning consignment results of Jan 2019 to consignors.
* The bottom chart shows the **Usable Budget** and **Client Asset** overtime. It shows that, the usable budget gradually increases and there are 3 drops which corresponding with a lot of expense made in date 28th each month. However, the **Client Asset** is decreased only in the returning consignment results period.
* Click **Backup** button to save all data to xml files in application folder **BackupData**.

# **5. Project Implementation**

## 5.1 Seed Data

Folder: SeedData

Include: ConsignmentStoreSeedData class and 3 insert data sql files for consignors, employees and items (generated from mockaroo.com website).

When application is started, if there is no data then ConsignmentStoreSeedData will insert seed data to all tables.

Process:

1. Insert 1 admin and 9 employees to Employees table
2. Insert 100 consignors to Consignors table
3. Insert 3,800 items to Items table (Jan/2019: 1000 items; Feb/2019: 1100 items; Mar/2019: 1200 items; Apr/1/2019 - Apr/7/2019: 500 items). Note that, all items in Jan/2019 were randomly set status as (“**sold**”, “**returned**” or “**lost**”) for simulate generate consignment results and summary for Jan 2019. The other items have status “**sold**” or “**unsold**”.
4. Insert soldItems to SoldItems table
   * Select all “**sold**” items and get their Ids
   * Randomly pick employee 1-10
   * Randomly pick SoldDate in range ConsignedDate - (ConsignedDate + 2)
   * 20% can get DiscountAmount $5
5. Generate a ConsignmentSummary of Jan 2019
6. Generate ConsignmentResults of Jan 2019 for related consignors
7. Generate some Expense Categories
   * Generate Discount Expense based on SoldItems table
   * Generate ConsignmentResult Expense based on ConsignmentResults table
   * Generate some monthly expenses at date 28th/Jan, Feb, Mar 2019
8. Generate Overview data for each day from Dec 31, 2018 to now

* Starting Budget is $5,000

## 5.2 Program Structure

|  |  |
| --- | --- |
| **File/Folder** | **Description** |
| **ConsignmentStoreDB folder** | Database project which use to create database for the application |
| **ConsigmentStoreApp folder** | Main application WinForm project folder |
| **ConsigmentStoreAppForm.cs** | Main Form app |
| **BackupDate folder** | Contains all generated xml files from database when clicking **Backup**  button in **Reports and Backup** screen |
| **SeedData folder** | Contains **CosignmentStoreSeedData** class and 3 sql files that used to seed the application database |
| **Controller/**  **ConsignmentStoreBusinessLogic.cs** | Provide database context for all the app, store logged in employee info, store some important business logic |
| **Utilities/FormHelper.cs** | Provide some useful methods such as Show Dialog Message |
| **UserControls folders** | Contains all UserControl that will be docked to main panel when corresponding button is clicked. Each UserControl contains its own logic to handle its functions. If the logic is use accross UserControl it will be implemented in **ConsignmentStoreBusinessLogic.cs** |
| **Forms folder** | Contain Forms that are used as Dialog (e.g. **LoginForm** is used by **ConsigmentStoreAppForm**, **AddEditConsignorForm** is used by UserControlManageConsignors) |

## 5.3 Program Setup

1. Publish ConsignmentStoreDB/ConsignmentStoreDB.publish.xml to generate database and all tables.
2. Run the **ConsignmentStoreApp** (it is slow at the first time because of data seeding).
3. Log in to the app (admin email: admin@consign.com; employee email: abra12@consign.com, ellene96@consign.com and all password is 123).