Class description

|  |  |  |
| --- | --- | --- |
| Folder | Class | Description |
|  | Authorization | Class containing user credentials stored in a dictionary |
|  | Employee | Class describing user properties |
|  | OrderCalc | Class containing methods with calculations related to order's cost, capacity, maximum distance that it can be carried, etc |
|  | ProductDelivery | Class describing product properties |
|  | Repository | Class containing method that builds path to a specified folder and text file |
|  | Route | Class (static) containing route specific method to lookup distance between departure and destination city. |
|  | StoreWarehouse | Class describing warehouse department and section division logic |
|  | User | Class describing user credentials. |
| ProductDealers | ProductIkea | Class for Ikea product dealer |
|  | ProductNestle | Class for Nestle product dealer |
|  | ProductNike | Class for Nike product dealer |
|  | ProductZara | Class for Zara product dealer |
| Delivery | CargoShip | Class contains cargo delivery description |
|  | DeliveryService | Class (abstract) contains delivery description |
|  | Drone | Class contains drone delivery description |
|  | Plane | Class contains plane delivery description |
|  | SantaClauseSledge | Class contains magic delivery description |
|  | Truck | Class contains land delivery description |
| Users | Administrator | Class describing different types of users |
|  | Partner | Class describing different types of users |
|  | WarehouseAdministrator | Class describing different types of users |
| Forms |  | Class describing different GUI forms, for details see user case below |

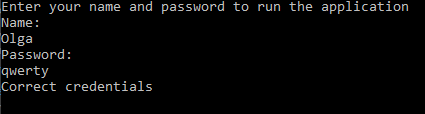
Business logic description

Pay Money Delivery Sir company provides delivery services. For this matter a Pay Money Sir delivery app was built covering company’s daily operations.

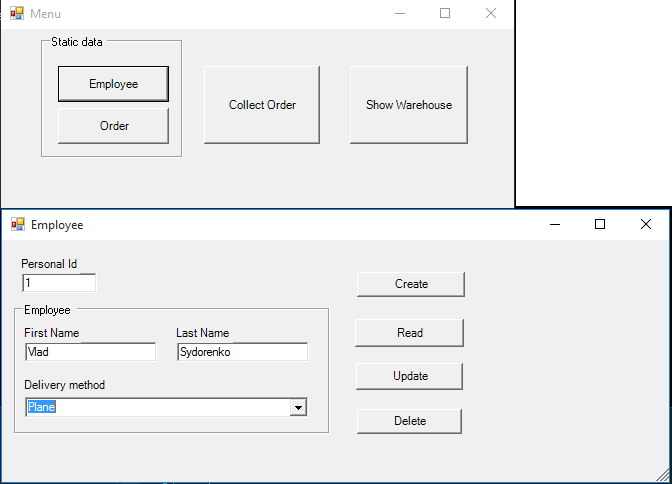
Employees can be enrolled according to the delivery method he/she can handle. Incoming orders are collected by each employee and are placed in the warehouse. Warehouse is divided by departments that consist of sections. Orders from warehouse may be pushed/pulled to/from conveyor. Search is available by department/section property or by just order’s Id.

User case for test

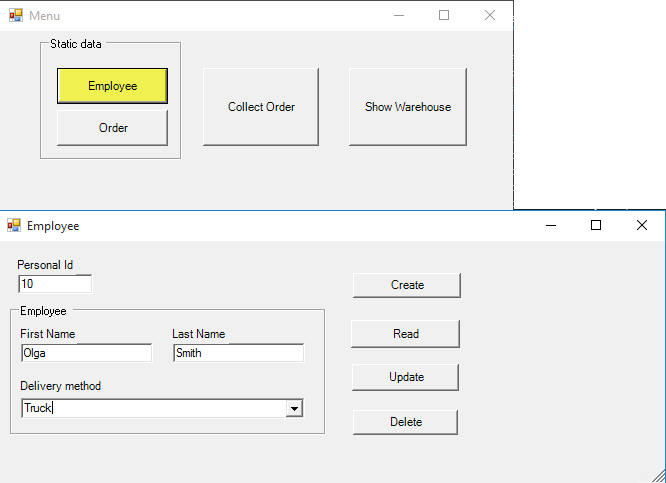
First of all user must log in using its validated credentials



Correctly entered credential grant an access to Pay Money Sir delivery application.



1 ) **Employee** form is used to Create/Read/Update/Delete information about employees in the company.



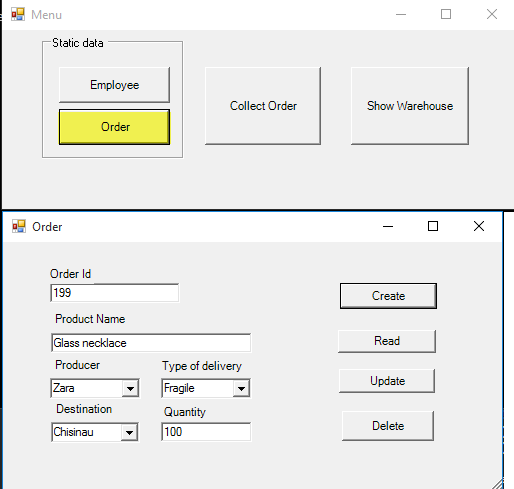
Create – new record is saved to “…\Employees\{PersonalId}.txt”;

Read – existing record information is loaded based on the ‘Personal ID’ value;

Update – existing record information is updated with values on the screen based on the ‘Personal ID’ value;

Delete – record is deleted based on the ‘Personal ID’ value.

2) **Order** form is used to Create/Read/Update/Delete information about incoming orders in the company.



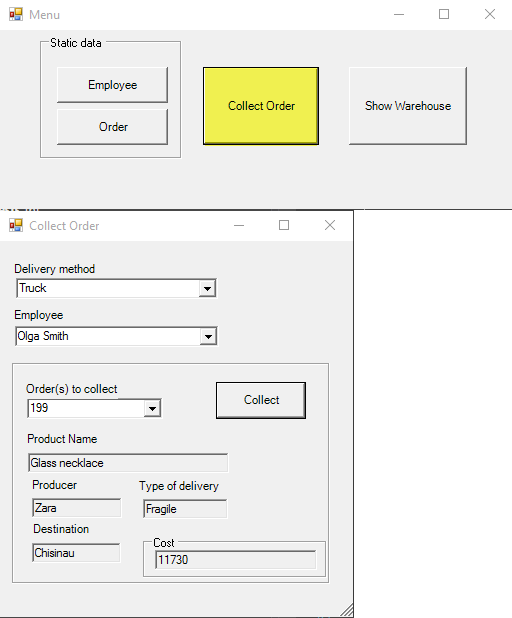
Create – new record is saved to “…\Orders\{OrderId}.txt”;

Read – existing record information is loaded based on the ‘Personal ID’ value;

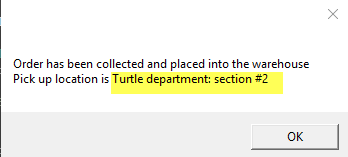
Update – existing record information is updated with values on the screen based on the ‘Personal ID’ value;

Delete – record is deleted based on the ‘Personal ID’ value.

3) **Collect Order**” form is used to collect orders by employees’ chosen method of delivery based on the destination, quantity and available routes. Collected orders are moved from “…\Orders\{OrderId}.txt” to “…\Warehouse\{OrderId}.txt” file repository.

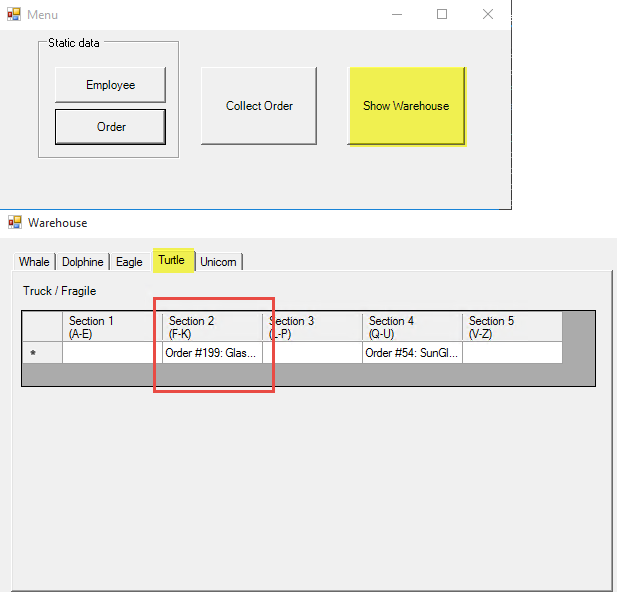


Collect – collect order and send it to a specific department and section in the warehouse.

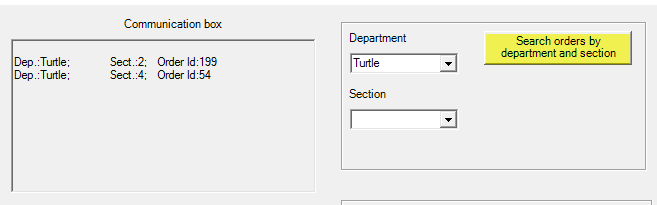


4) **Warehouse** form can be used in a multifunctional way:

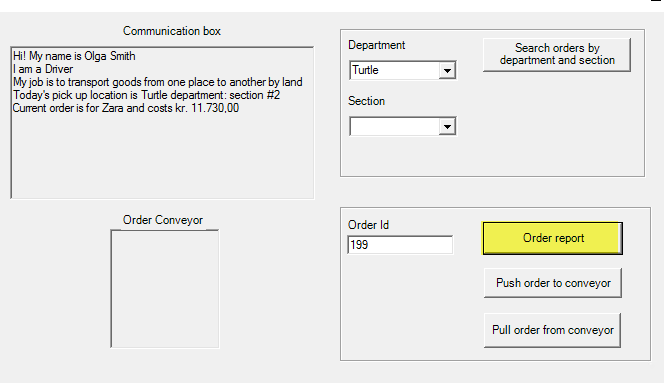
* View collected orders stored in departments and sections



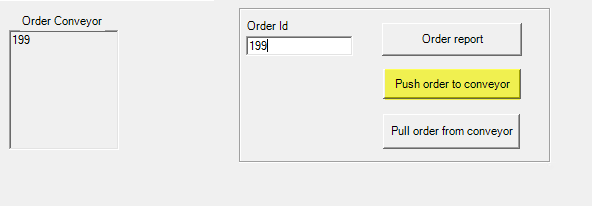
* Search for collected orders by departments and/or section. Results of the search is shown in the Communication box



* generate a report for a specific order. Report is shown in the Communication box.



* push/pull collected orders based on Order Id to/from conveyor.



Note: conveyor can hold up to 5 orders at once. To free space – pull order based on Order Id from the queue