1. Café table manager

2. The idea of the project is to develop an application that implements café management system for waiters.
   The application displays the plan of a dining hall and orders of a table when the latter is picked. It also calculates some general statistics on tables' and waiters' profit as well as the orders.

3. https://github.com/GoodBadSharp/TableManager

4. Minasyan Tigran - database design

   Makrinova Yulia - UI application and logic

   Ivashneva Ekaterina - application logic, paperwork, data seeding

5. Interfaces

   a. IOrdersRepository //Provides actions implied to orders, works with database
   b. Iqueries //Stores main data about queries and supplies the actions with queries
   c. ITablesRepository //Stores operations for any table interaction

   Model

a. Dish
b. DishInOrder
c. Order
d. OrderStatus
e. Table

f. TableStatus
g. Waiter

Model Classes represent entities of the Database with its properties, including DishInOrder that is a staging table for the 'many-to-many' relationship of Dishes and Orders.

QueryLogic

a. Queries //Main logic for queries
b. QueryContainer //Passes the data about available queries to the UI segment
c. QueryResult //Incapsulates queries' results and passes them to the UI segment

Repositories

a. OrdersRepository //Stores easily accessible data about orders
b. TablesRepository // Implements actions aimed at tables

Context

Entity set for all entities and relationship-setting constructor

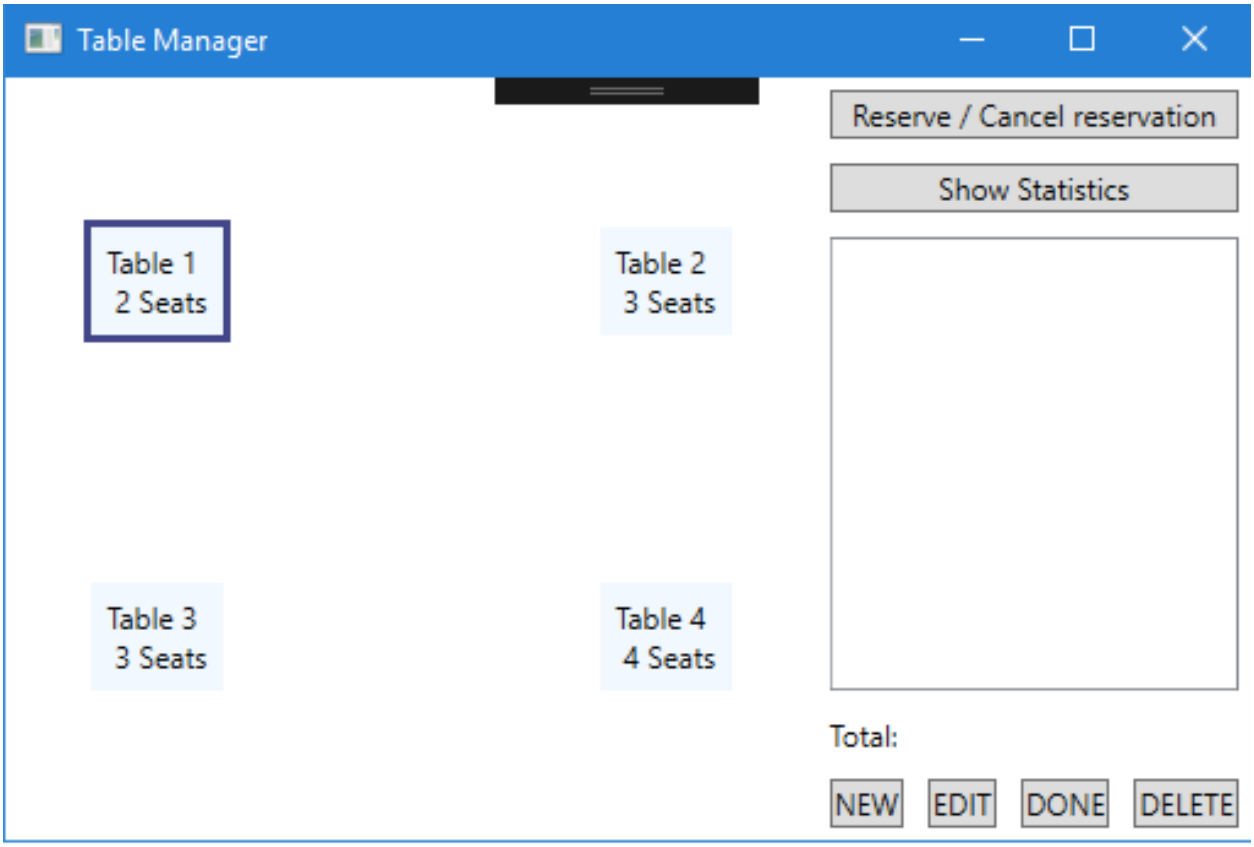
UnitOfWork

Keeps track of all the possible changes to the database and males sure they are saved

PageContainer

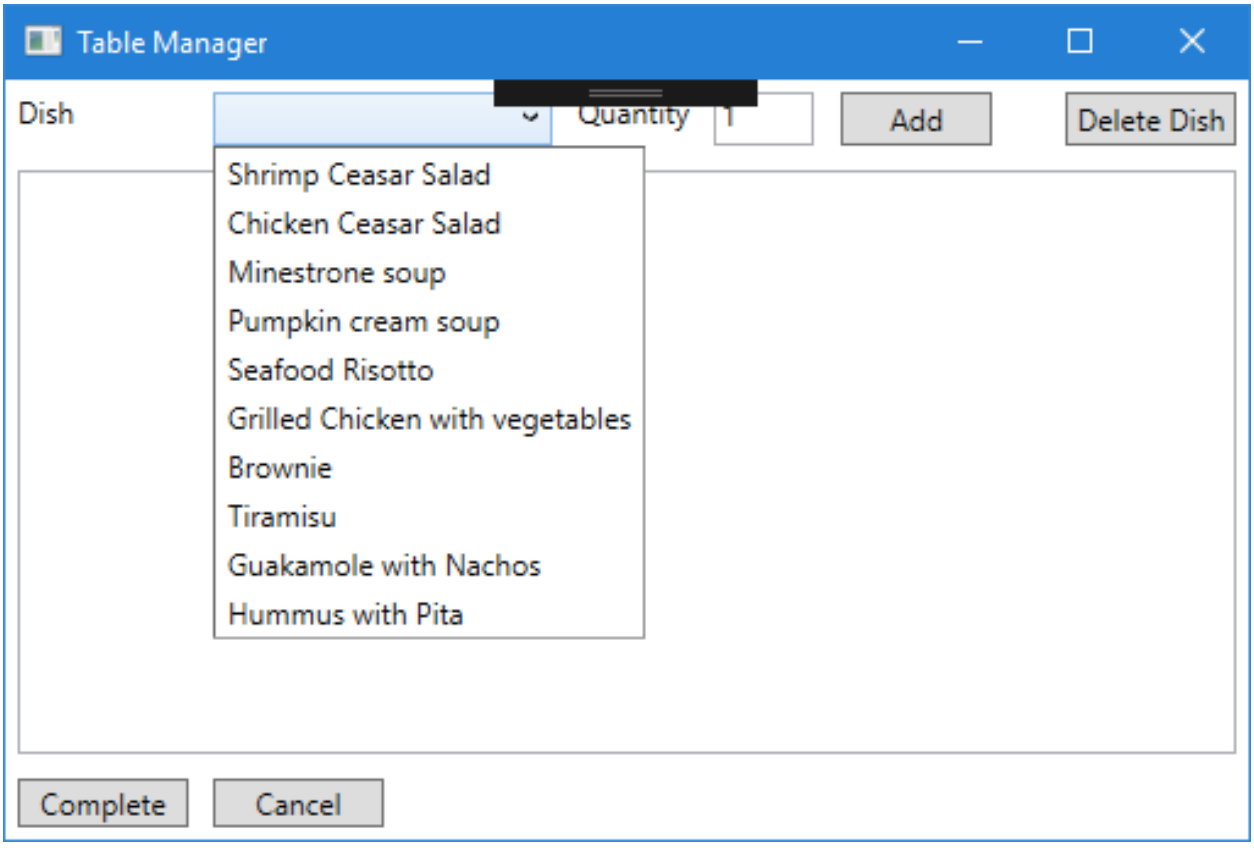
Contains static instances of the program's all pages

6. (Overview)The program displays the plan of a dinning hall as a grid of tables with its unique identification numbers for waiters to navigate easily. The status of each table (vacant, reserved or occupied) is visible in the overview as well as the number of seats. Orders assigned to the chosen table are showed in a tree view on the right.

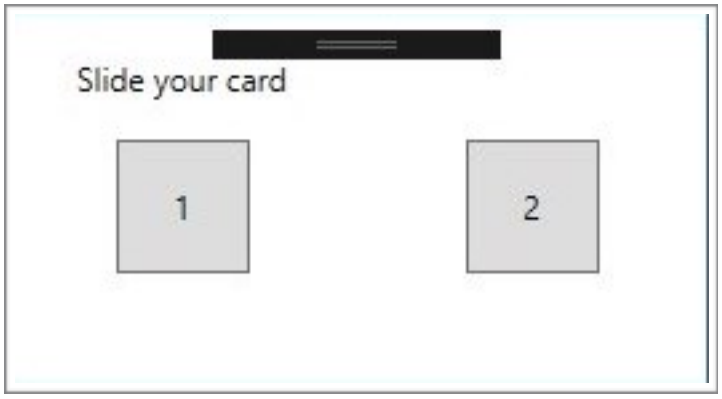


(Add order) List of all dishes available for ordering and the corresponding button. User enters the quantity of a certain dish.

Dishes of the order are displayed. After everything is added, the order should be completed.

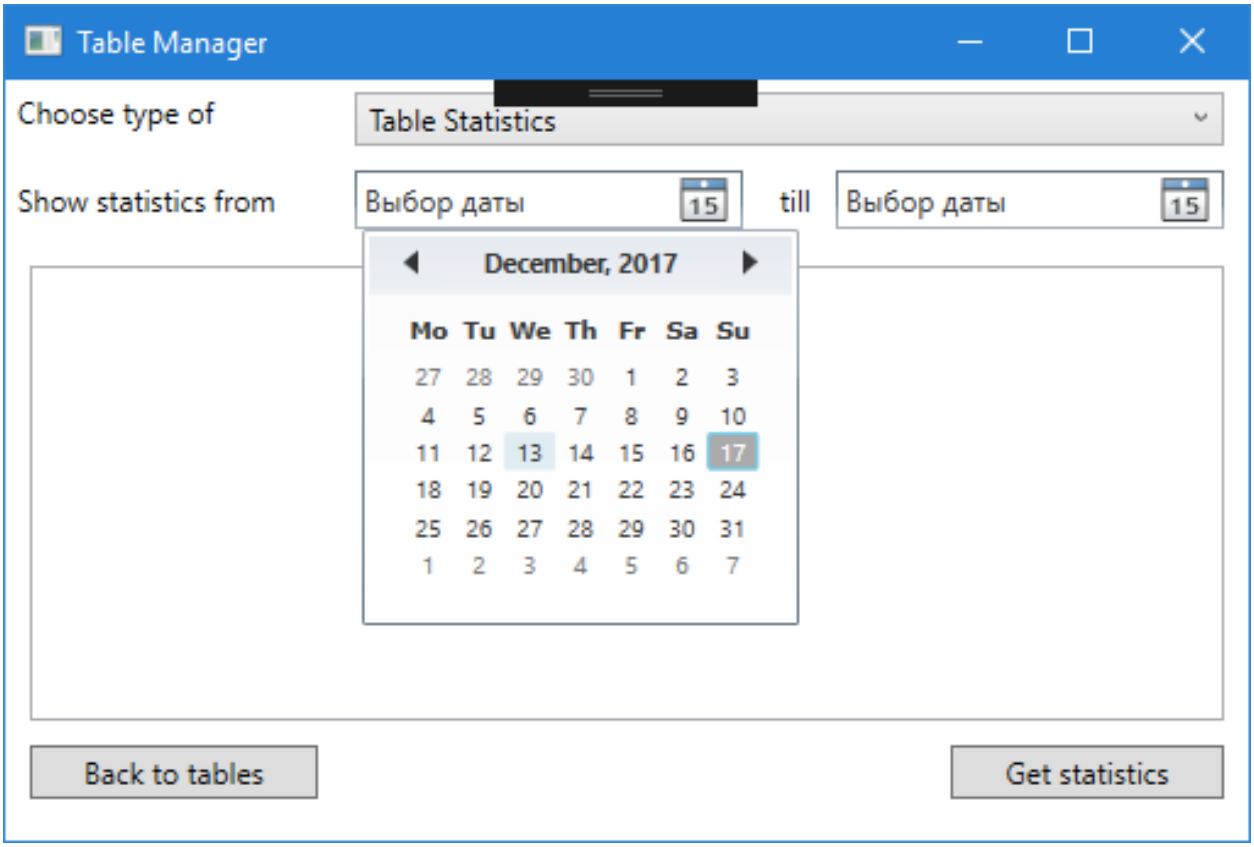


(WaiterAuthorization)
Buttons correspond to waiter's identification number, one has to chose his and proceed to the main page. The window emulates sliding Id card in the R-Keeper

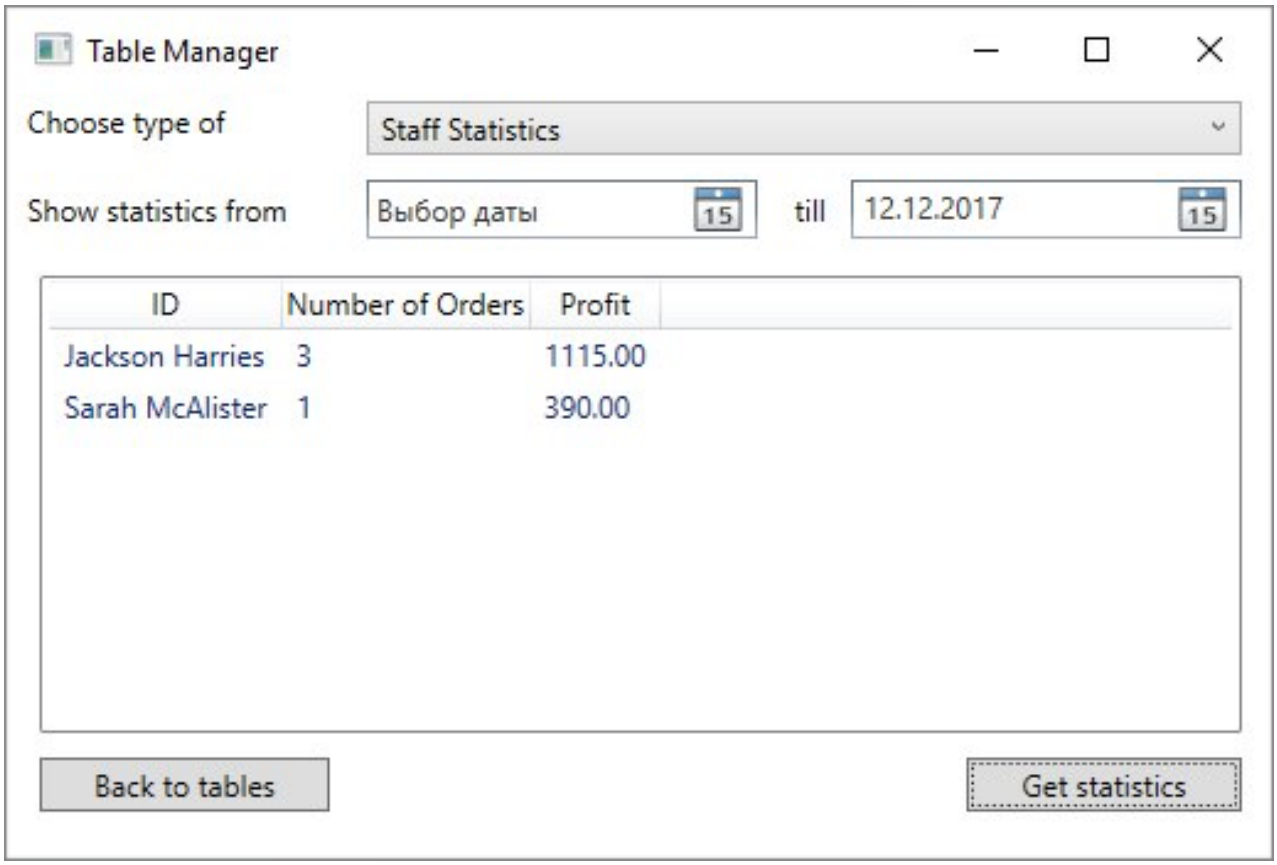


(Statistics) Table Statistics and Staff Statistics

The Statistics can be calculated either for tables or for staff members, which is chosen in the combo box on top. Afterwards the period of the needed report has to be set. If any of the dates is not stated, the program sets the earliest and the latest date possible respectively by default.

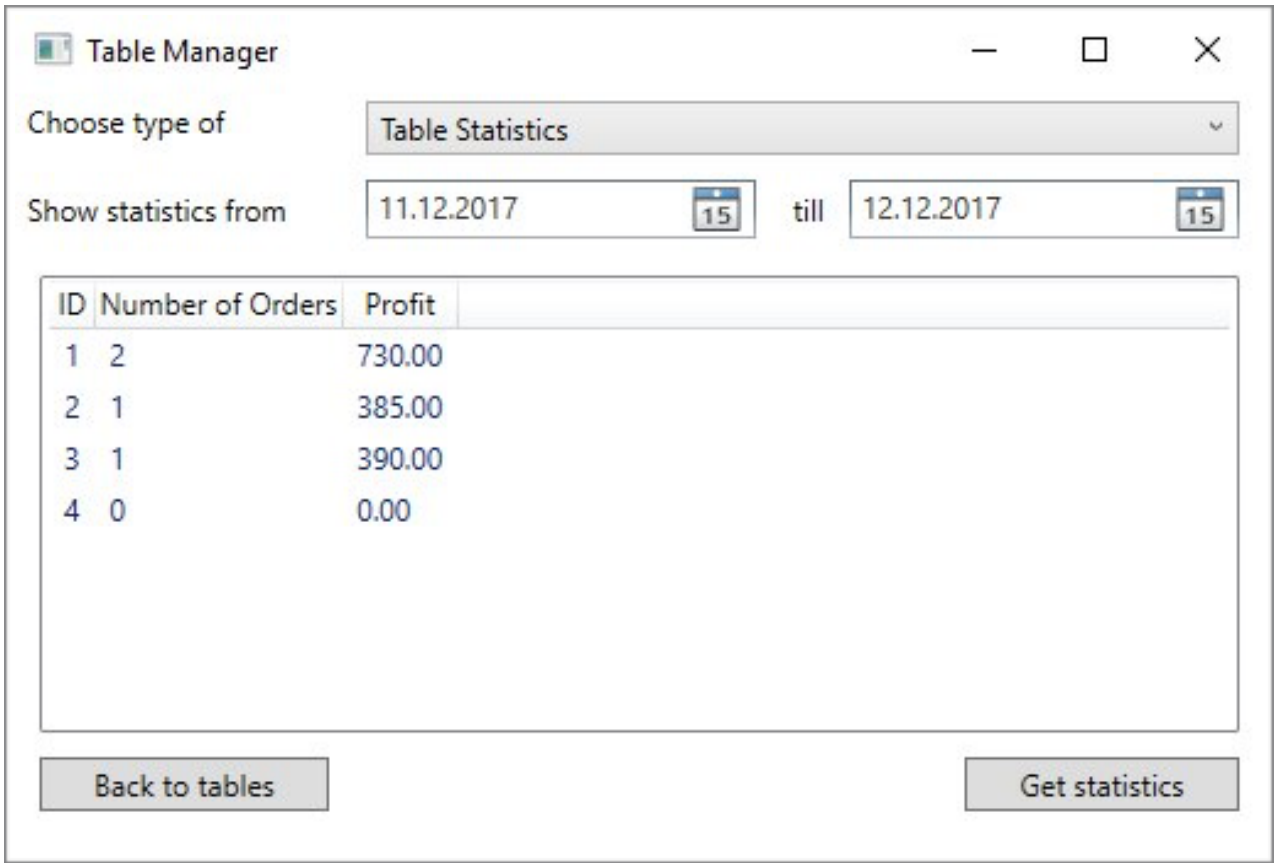


7. Testing Case 1



Getting the statistic data for all the staff members. Currently a café has 2 waiters. User wants to get the accumulated profit of each waiter until December 12. We do not choose the start date as we want to include all orders serviced by our staff, therefore the first date is set by our program by default. In the report we get the Name of a waiter, the number of orders he has completed and the total profit he has made.

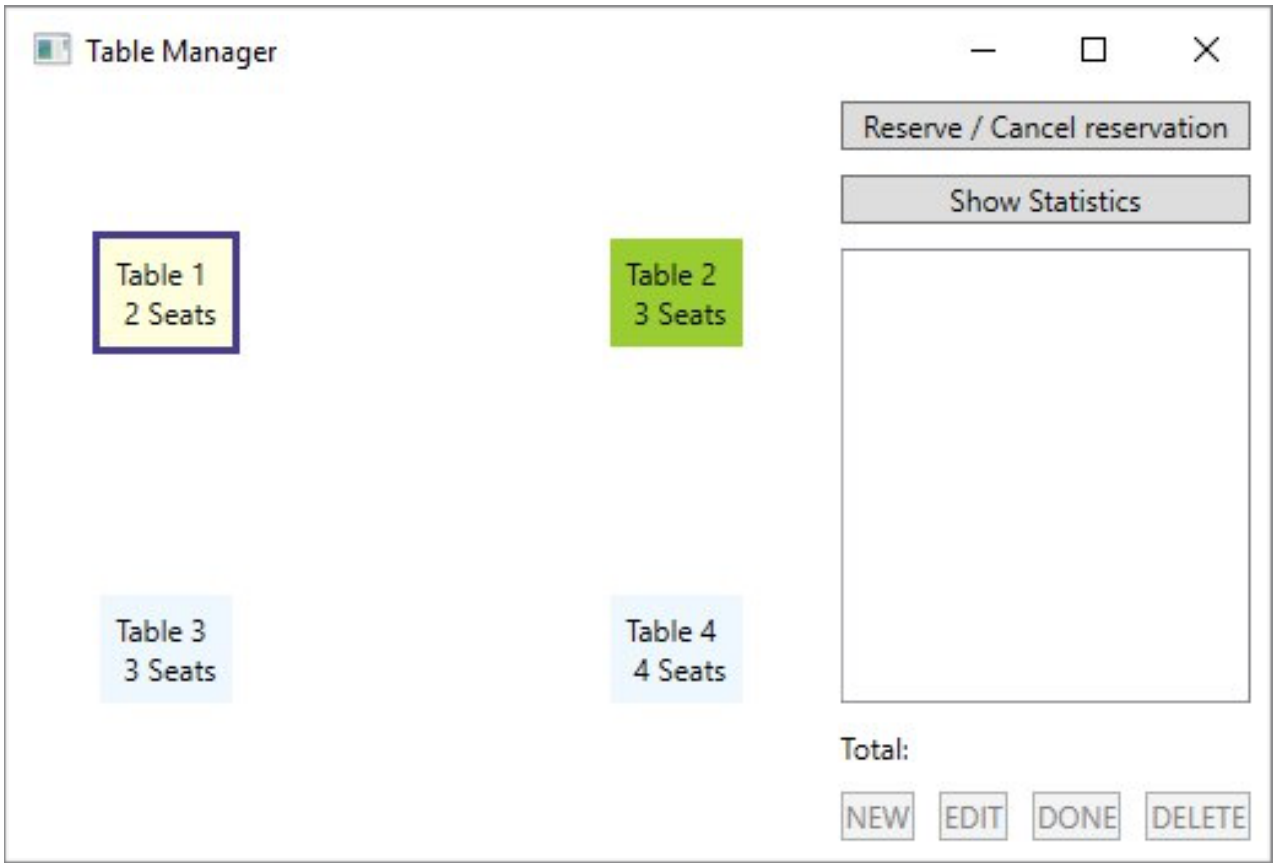
Testing Case 2



We want to get the statistics for each table from December 11th till December 12th. What we get in the report is the list of all tables of the café with the corresponding accumulated profit over the period.

Testing Case 3

Customer wants to reserve a table. On the main page the user sees the plan of the café, among all the vacant tables (greyish-blue) he can tap one (dark border appears for the chosen table) and press button 'Reserve/Cancel Reservation', the table becomes yellow



meaning it is reserved. If a reservation needs to be cancelled, the same button is pushes and the table becomes neutral in color again, its' status changed back to vacant.

Testing Case 4

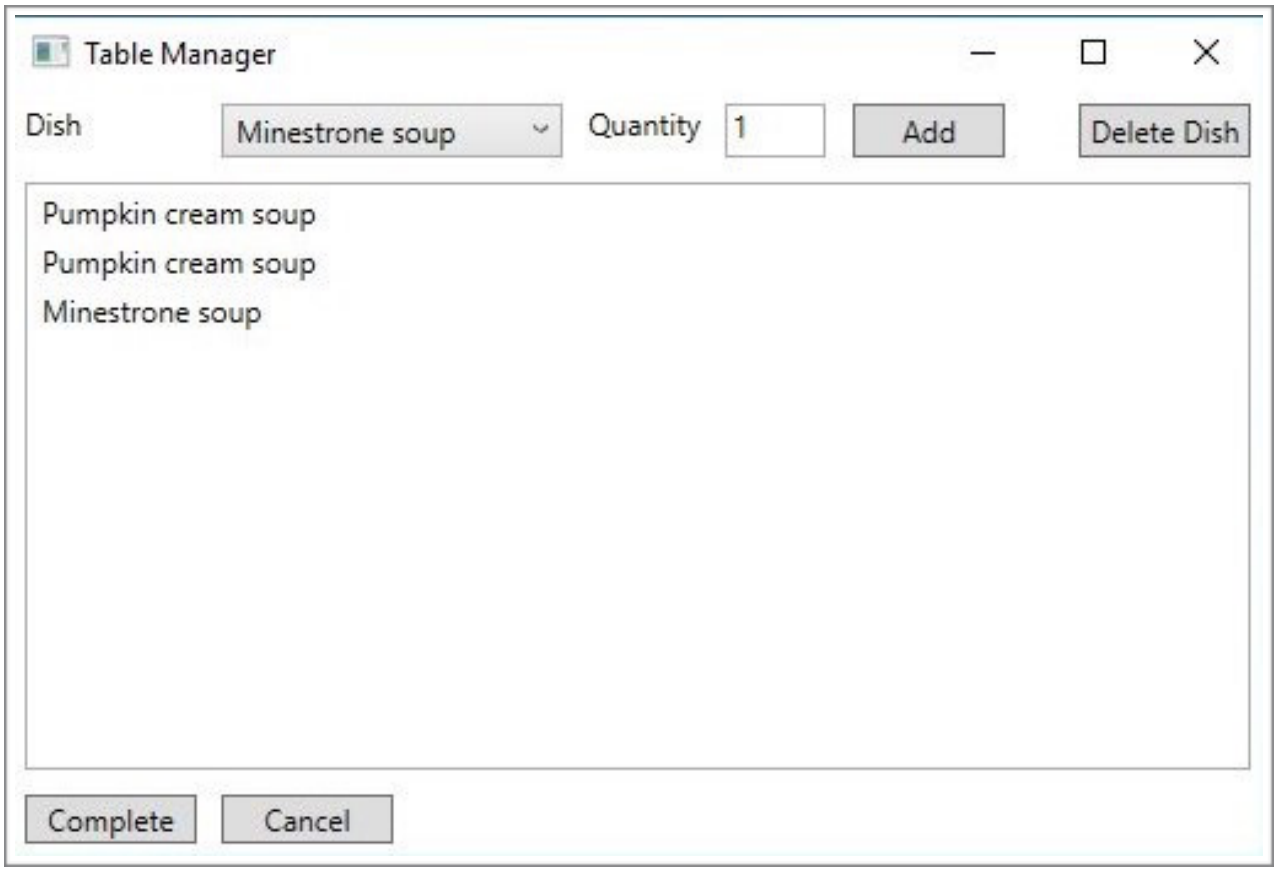


Table Manager
Dish
Minestrone soup
Quantity
1
Add
Delete Dish
Pumpkin cream soup
Pumpkin cream soup
Minestrone soup
Complete
Cancel

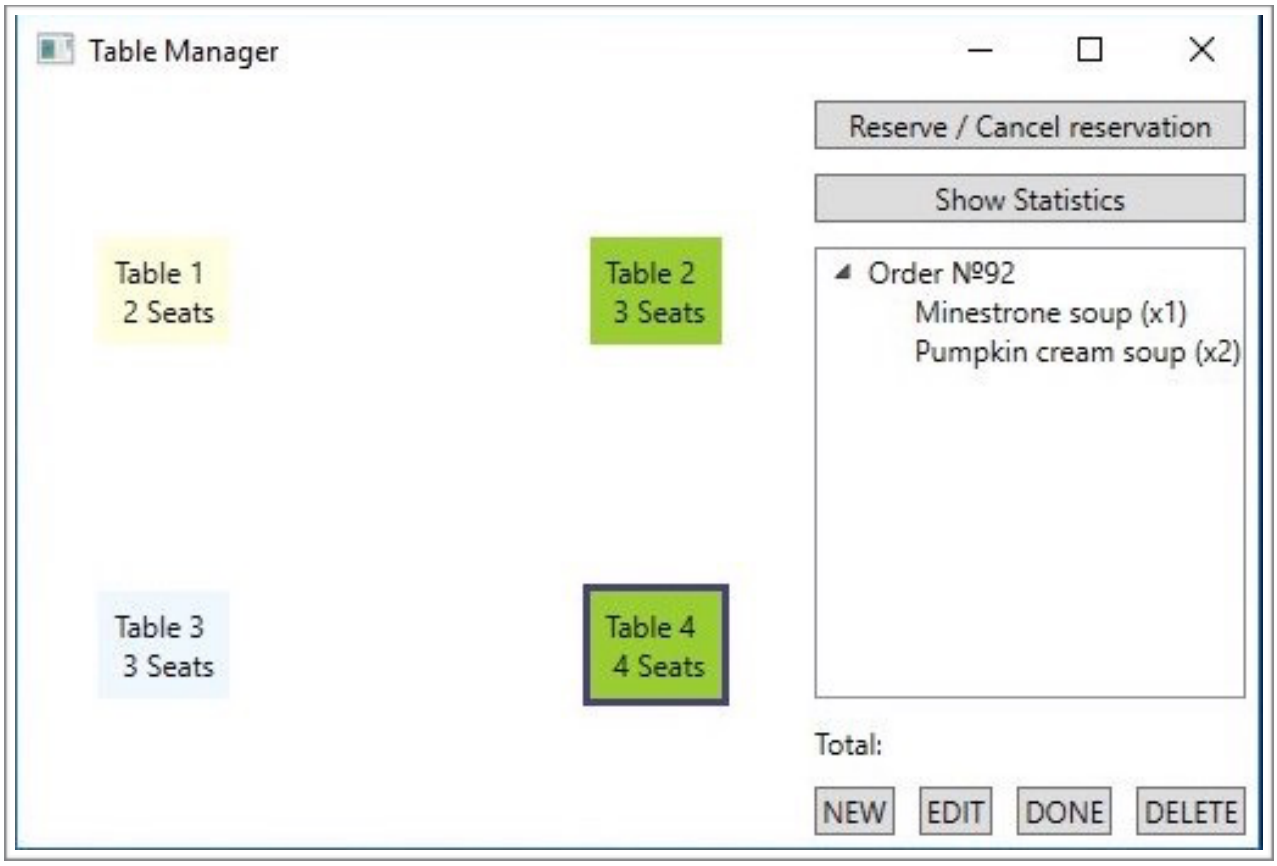
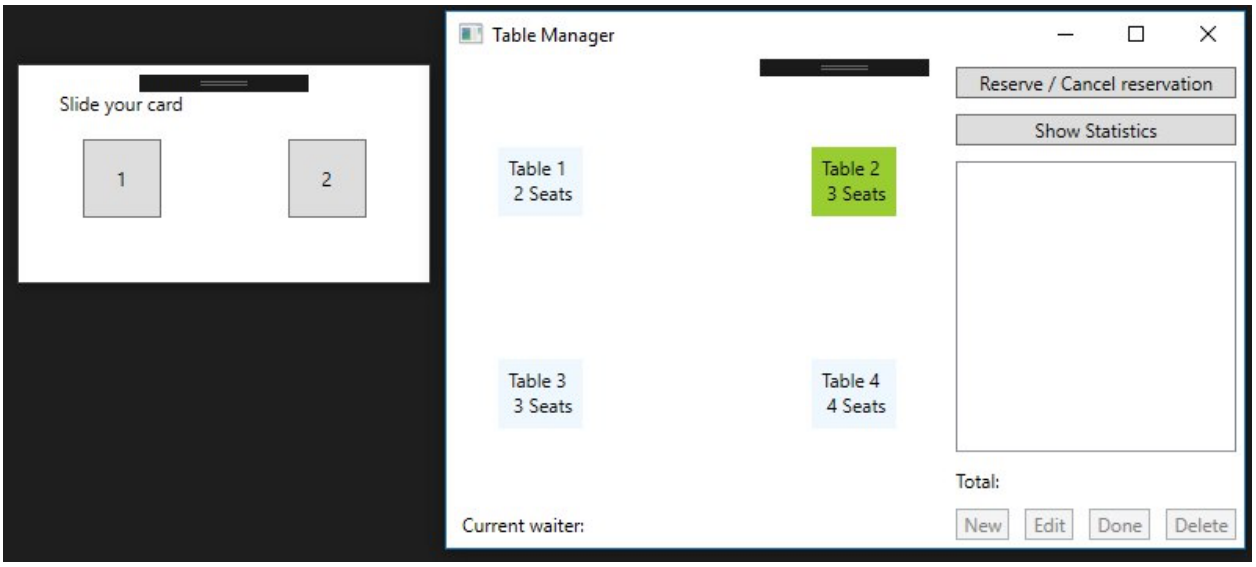


Table Manager
Reserve / Cancel reservation
Show Statistics
Table 1
2 Seats
Table 2
3 Seats
Order №92
Minestrone soup (x1)
Pumpkin cream soup (x2)
Table 3
3 Seats
Table 4
4 Seats
Total:
NEW
EDIT
DONE
DELETE

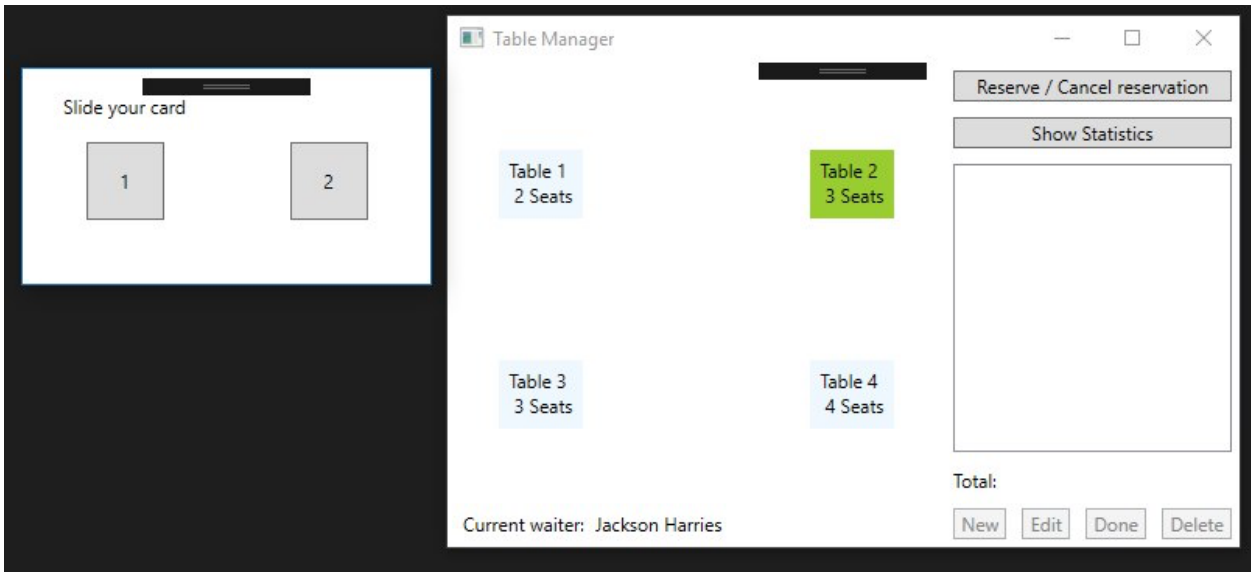
The user wants to form a new order. When clicking 'New' he is transferred to the Adding Order page, where he chooses the dishes and their quantity. After all data corresponds correctly to the actual order, the user presses 'Complete', which navigates him back to the main page.

Now when a table is selected, the order assigned to it is displayed in a tree view on the right. The preview includes the name of the dish and the quantity.

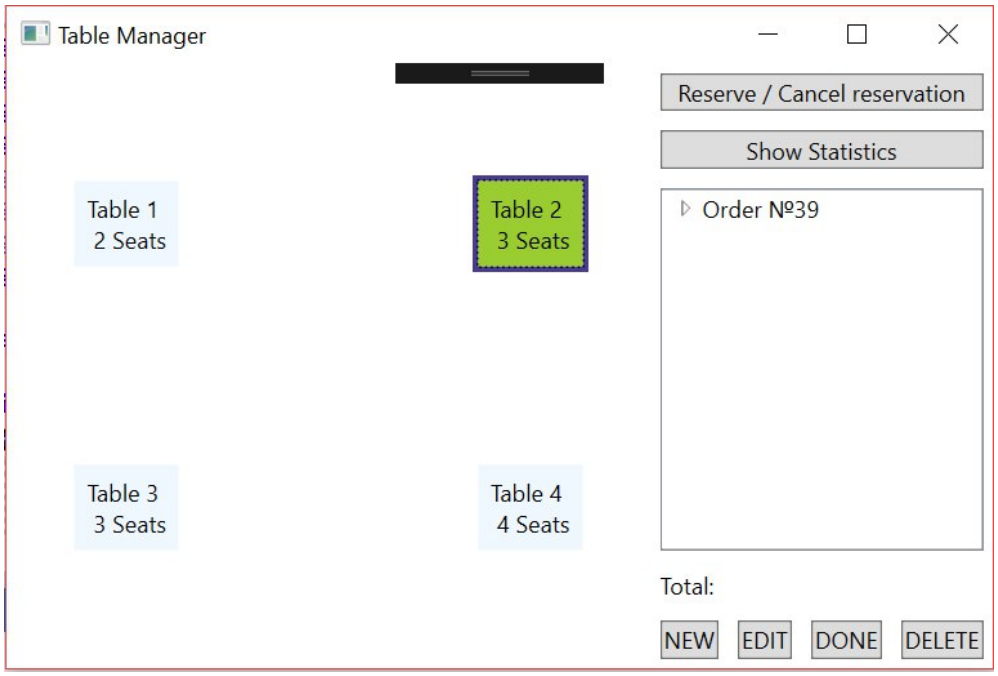
Testing Case 5



Unauthorized user wants to add order. He is unable to do so, but he can see the plan of a cafe. The user then log ins by pressing the corresponding button. Now waiter's name is displayed in the bottom and he can add orders.

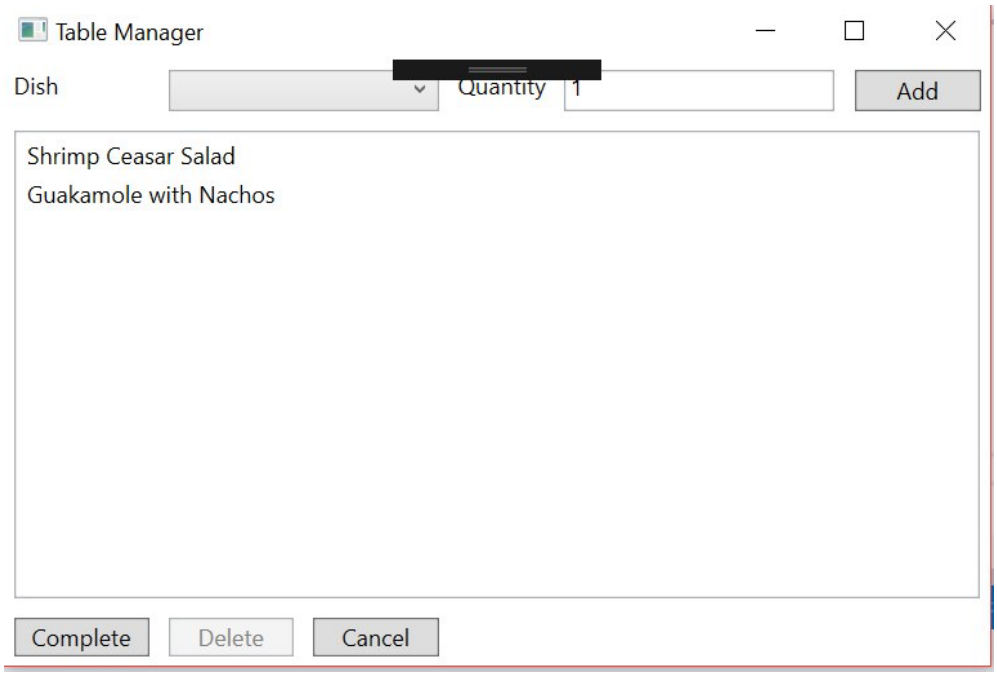


Testing Case 6

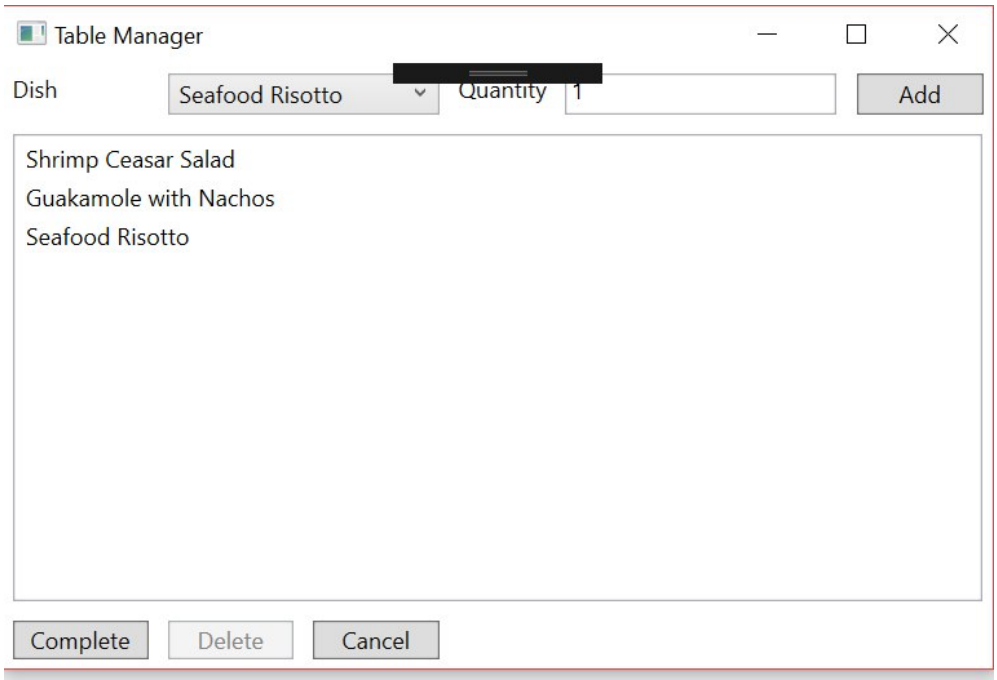


An order with two dishes is assigned to a table.

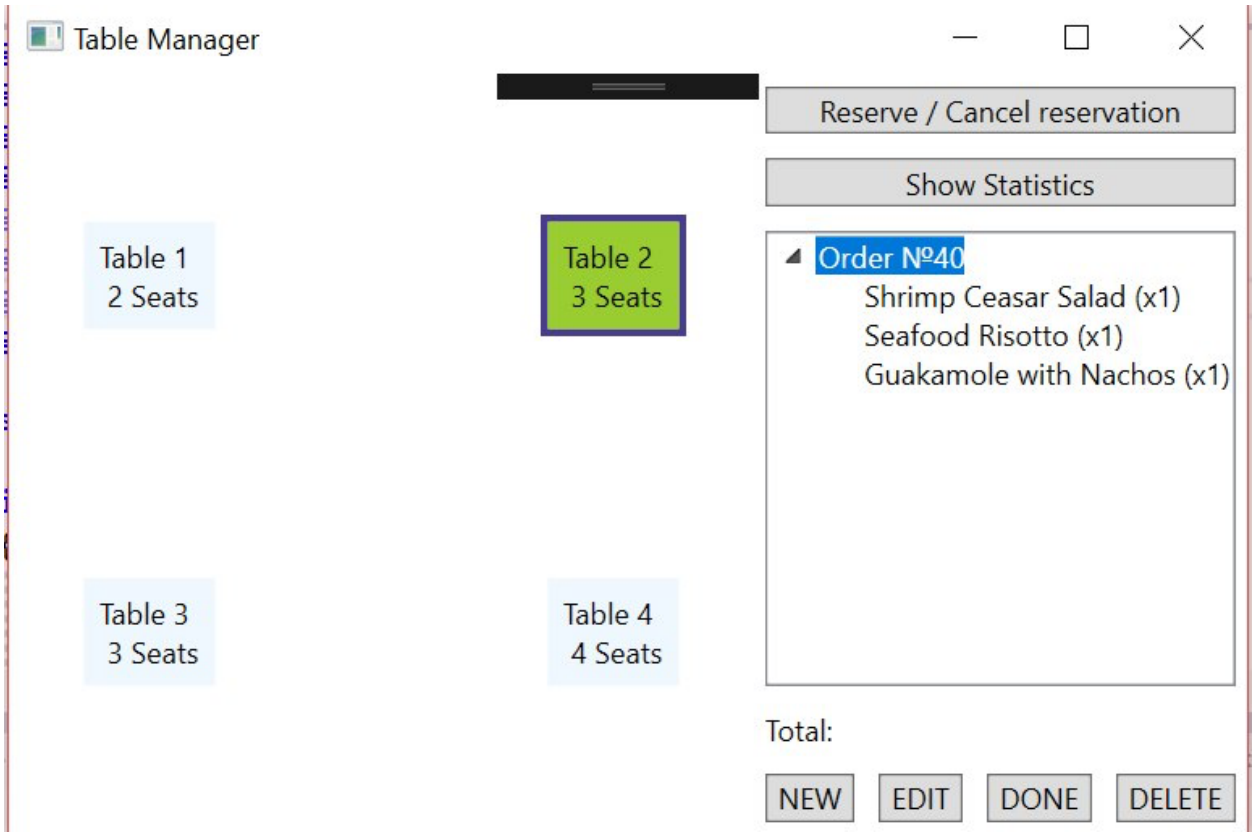
Now we want to add another dish to it. ‘Edit’ has to be pressed.



An editing window opens, which displays dishes that are in the order. By our standard procedure we choose the dish we want to add and press ‘Complete’ button.



Now three dishes are assigned to the order



On the main page an order assigned to the table is now displayed with added dishes.