# The Restaurant Orders Management - ROM

## Introduction

* 1. The Restaurant Order Management - ROM (henceforth called the system) is a software application to assist handling restaurant customers’ orders.
  2. The main advantages of the system are:
     1. Reduce effort for working under pressure waiters while taking orders from customers.
     2. Support various payment options.
     3. Be easy and simple to use.

# Overall Description

* 1. Create a user friendly GUI for working with ROM.
  2. Taking order from costumers.
  3. Interface with eatWithIt app.
  4. Collect statistics about costumers ordering behaviors, favorite meal and so on.

# Functional Requirements

* 1. The menu – a list of meals
     1. When the user asks, the system shall add a new menu category.
     2. When the user asks, the system shall edit a menu category name.
     3. When the user asks, the system shall add a new meal to a specific category with the following attributes - name, description, toppings and price.
        1. For each topping, a different price will be displayed.
     4. When the user asks, the system shall edit/delete a topping and its price from a specific meal.
     5. When the user asks, the system shall edit/delete a meal from the menu.
     6. When the user asks, the system shall move a meal from one category to another.
     7. When the user asks, the system shall delete an empty category.
        1. Before deleting a category, all meals must be deleted or moved to other categories.
  2. Take orders from costumers
     1. When a table of customers asks for a menu, the system shall display the restaurant menu in such the customers can choose the meals from it.
     2. After selecting all meals, the system shall pass the information to the relevant section (kitchen, bar or making-cakes stand).
        1. A table of customers can change it choice while the meals are not ready. The system shall update the meals list (a dinner) while the meals are not ready.
        2. When a relevant section cannot make more a specific meal, the system will display that the meal is unavailable.
     3. The system shall display a message to the waiters when the dinner is ready to take.
     4. The system shall print the final bill when the dinner is done.
     5. The system shall display three options for a payment – cash, credit or eatWithIt app.
        1. The payment can be for each customer or for the whole table.
     6. The system shall print the reception when either cash or credit payment was chosen.
  3. Collect statistics of customers’ orders
     1. The system shall collect and display statistics for the following attributes:
        1. The most ordered meal in each category.
        2. The least ordered meal in each category.
        3. An average price for a bill in each day of the week.
        4. An average price for a customer in each day of the week.
        5. The most favorite payment choice (cash, credit or eatWithIt) in between range of prices in each day of the week.

# Non-Functional Requirements

* 1. Performance
     1. The system shall be simple as possible and easy for manufacture.
     2. The system shall be user friendly for the waiters, customers and the restaurant manager.
     3. The system shall be synchronized with the waiters’ hardware and the sectors’ hardware for passing information about the orders statuses.
  2. Portability
     1. The system will be implemented on a platform that allows easy re-hosting on different hardware and operating system.

# Future Requirement

* 1. Support games/apps for the customers while having the system and waiting for a waiter.