Restaurant 3.0, the big Mc. . . era

Hello again. This time you may use whatever fancy stuff the executor framework has to offer. The game is to get as many meals ordered, cooked and served in the least amount of time.

The hard part is in proper shutdown of the application. The customer will be very annoyed if not all ordered (and payed) meals have be served before the joint is closed.

Some hints:

- Consider refactoring all work into tiny independent tasks, such as order, cook and serve tasks and feed them to a properly chooses and configured executor.
- Consider using a combination of *poison pills* and other ways to shutdown.
- Consider not just an empty order queue (which may happen more often during opening hours, when business is (s)low) but also regular closing hours. Mind that no order may be left unserved.
- Maybe using a priority queue as the queue to stick you tasks in. The choice and implementation of the priority rules is of big importance

This task is an extension of the previous restaurant task.

Before you start, make a TAG of your second restaurant using the appropriate subversion command. Use the TAG `.../tags/RestaurantV2`.

Document everything in a report and hand the report in on paper. The report should contain the group number and names and numbers of the students of the group as well as the names of the authors of the report.

1. Document you improvements as well. Hand in your results in print, before the due time as mentioned on the website.
2. Spend some remarks on scalability and boundness of the application.