**Restaurant Automation — Solution**

Brief use case descriptions are as follows.

UC1: *ClockIn* — Employee records the start time of his/her shift or upon arriving back from a lunch break, assuming, of course, that the employee clocked-out before going to lunch.

UC2: *ClockOut* — Employee records the end time of his/her shift or when going out for a lunch break. (The system could automatically log out the employee from any open sessions.)

UC3: *LogIn* — Employee logs-in to the system in order to perform his/her necessary functions.

UC4: *LogOut* — Employee logs-out of the system, including if another employee needs to use that terminal.

UC5: *MarkTableReady* — Busboy marks a table ready for use after it has been cleaned and prepared for a new party. The Host is automatically notified, so now they can seat a new customer party at this table.

UC6: *SeatTable* — Host seats a customer, marks the table as occupied and assigns a waiter to it.

UC7: *AddItem* — Waiter adds an item to a table’s tab.

UC8: *RemoveItem* — Waiter removes an item from a table’s tab that does not belong there. The Manager enters his/her authorization code to complete the item removal process.

UC9: *AdjustPrice* — Waiter adjusts the price of a menu item due to a coupon, promotion, or customer dissatisfaction.

UC10: *ViewTab* — Waiter views the current tab of a particular table.

UC11: *CloseTab* — Waiter indicates that a tab has been paid, and that the transaction is completed. The table’s tab’s values are reset to “empty” or “0” but the transaction is recorded in the database. The system automatically notifies the Busboy so that he/she can clean the “dirty” table. (There could be an intermediate step to wait until the party leaves their table and only then the Waiter to register a table as waiting to be cleared.)

UC12: *PlaceOrder* — Waiter indicates that a table’s tab is completed. The kitchen staff (Cook) is notified that the order must be prepared.

UC13: *MarkOrderReady* — Cook announces the completion of an order. The status of the order tab is changed, the tab is removed from the order queue in the kitchen, and the appropriate Waiter is notified.

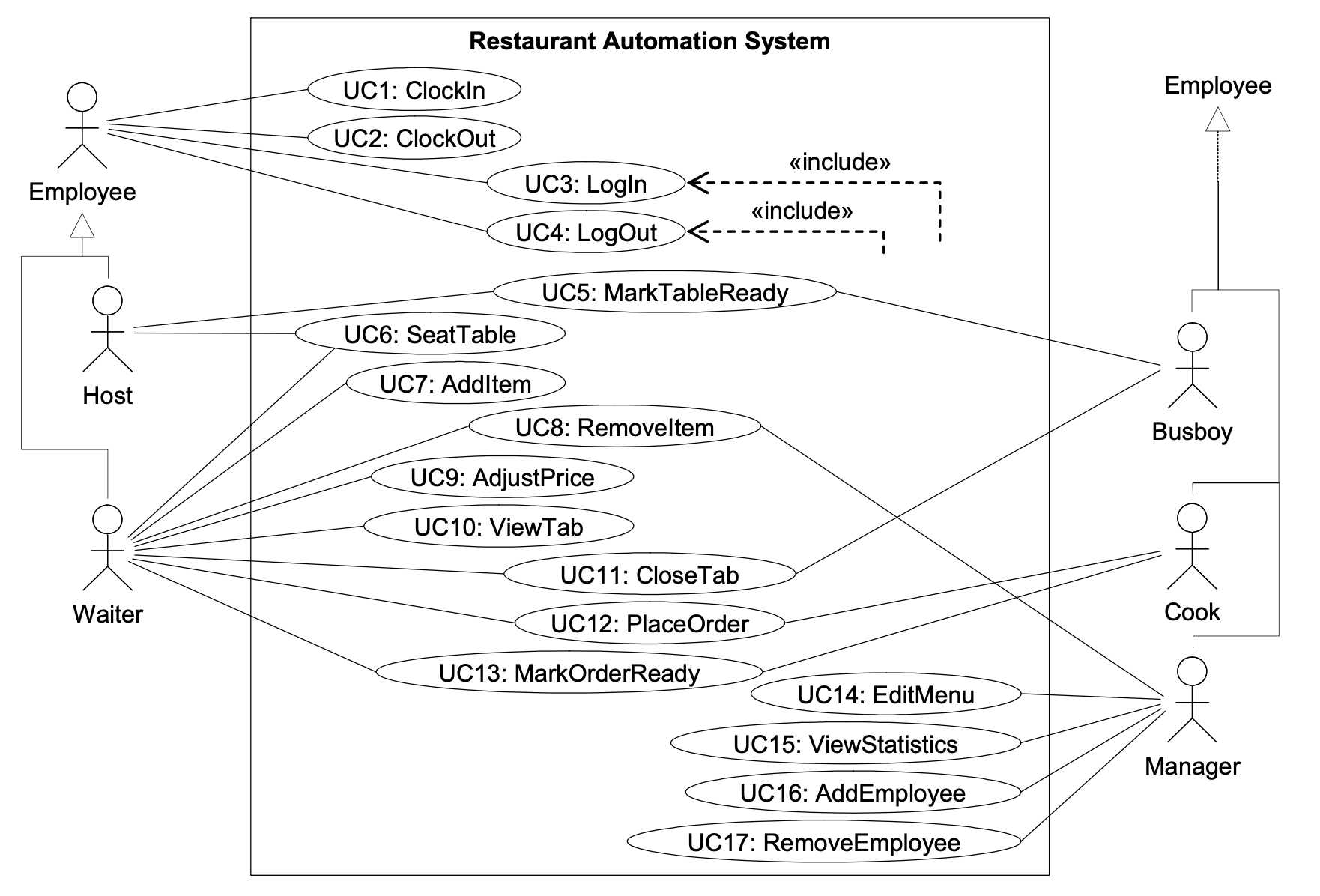
UC14: *EditMenu* — Manager modifies the parameters of a menu item (name, price, description, etc.) or add/removes an item to the menu.

UC15: *ViewStatistics* — Manager inspects the statistics of the restaurant.

UC16: *AddEmployee* — Manager creates a profile for a new employee. The profile will contain information pertinent to that employee, such as employee name, telephone number, ID, salary and position.

UC17: *RemoveEmployee* — Manager deletes a profile of a former employee.

The use case diagram is shown in Figure H-1. The auxiliary uses cases UC3 and UC4 for login/logout are included by other use case (except UC1 and UC2), but the lines are not drawn to avoid cluttering the diagram. There could also be a use case for the Manager to edit an existing employee profile when some of the parameters of an employee profile need to be changed. The reader should carefully trace the communications between the actors and the use cases and compare these with the brief description of the use cases, given above.



**Figure H-1: The use case diagram for the restaurant automation project.**

I chose to include the database as part of the system because I feel that showing it as an external system (supporting actor), although true, would not contribute much to the informativeness of the diagram.