

# Requirement specification based on Personas and Scenarios

## Note:

Items that are placed inside [...] should NOT be included in the application.  
Items placed within {...} can be included at will, but they are not necessary.

## Functional requirements

*The system should provide functionality that allows all customers to:*

1. See available items (open menu)
2. Order beverages {and food} from servicing bartender at the table
  - a. Single order
  - b. Group order
    - Group bill
    - Split bill
3. Change order
4. Pay at bar, or to bartender/waiter/waitress at table
  - a. See also 2.b.ii - splitting the bill
  - b. Transaction is not implemented more than symbolically
    - The new amount can simply be entered (no card reader, Swish or similar equipment should be implemented)

*VIP customers should also be able to:*

5. Log in (at table)
  - a. Simple login!
  - b. No specific requirements on security in this project.
6. Log out
7. {Order food}
8. See account balance (at table)
9. Order and Pay from account (at table)
10. Fetch special beer/drink from fridge or bar (with combination lock)
11. Add to account (at bar)

*Bartender, Waiter and Waitress should be able to:*

12. Log in
  - a. Simple login!
  - b. No specific requirements on security in this project.
13. Log out
14. See availability of product (General req)
  - a. Remove product (temporarily) from menu
15. {Modify/Recalculate price of product as compensation for mistake}. This is used in order to compensate customers, if we cannot offer it on the house (see 16).
  - a. {The Price modification has to be entered with a comment as to the reason.}
  - b.



*The Flying Dutchman – Nataly Grinchenko*

16. Offer product on the house
  - a. Balancing the income/expenses
  - b. Update number in stock
17. Notify security of problem
18. Get order for certain table
  - a. Change items on order
  - b. Allow for splitting the bill
19. {Reserve table for group at specified time}
  - a. {Remove reservation}
  - b. {Print reservation note}

*Manager/Owner needs to be able to:*

20. Manage stock
  - a. Revise amounts
  - b. Order refill of items
  - c. Add/remove items from menu
21. [Manage prices.]
22. [Do accounting.]

#### *General requirements*

23. We need to be able to find products according to content
  - a. Allergies - Gluten, Nuts, Lactose etc.
  - b. Alcohol content
  - c. Tannins (for wine)

## Non-functional requirements

24. An order can consist of up to ten items at the same time
25. The bartender should immediately be notified when an item is running low
  - a. A warning should be given when there are less than five items left of a certain type
26. The security notification should be accessible within three seconds (17)

#### *Additional details*

27. The interface must connect to the thematic background of the Pub
28. Drinks should be listed with the following details on the menu:
  - a. Beers
    - Name
    - Producer/Brewery
    - Country
    - Type (IPA, lager)
    - Strength
    - Serving size (tap, bottle)
    - Price
  - b. Wine
    - Name
    - Year
    - Producer
    - Type
    - Grape
    - Serving size (glass, bottle)
  - c. Cocktails/Drinks

- Name
  - Strength
  - Contents/Recipe (for allergy purposes)
  - Serving size
29. For the bartender and manager, the number of remaining servings should also be shown
- a. It should be easy to see when an item is low in number

#### Notes

30. There should be one 9"-10" touch-screen at each table, and a large screen at the bar.
31. Waiters and Bartender should also have a small tablet for taking orders at the tables

### Course based requirements

32. The system should be implemented using MVC
33. The system should provide three distinct interface languages
34. The system should implement Drag and Drop *for suitable actions*
35. The system should implement an UNDO/REDO functionality
- a. at least two different actions are required
  - b. unlimited undo-redo
36. The system should be possible to resize between the bartender view (27") to table tablet (9"-10")
37. The system should be reasonably debugged
38. The code needs to be well documented
39. In order to pass the course, you only need to be able to show that the system can run on one platform, on one browser.