DIT636 / DAT560 - Assignment 1: Quality Scenarios

Due Date: Sunday, February 8th, 23:59 (Via Canvas)

There are four questions worth a total of 100 points. You may discuss these problems in your teams and turn in a single submission for the team on Canvas. Answers must be original and not copied from online sources.

Cover Page: On the cover page of your assignment, include the name of the course, the date, your group number, and a list of your group members.

Peer Evaluation: All students must also submit a peer evaluation form. This is a separate, individual submission on Canvas.

Overview

You are developing an automation system for a restaurant. This system offers a library of core functionality for different client systems, both online and in the restaurant.

The client systems include:

- The website for the restaurant.
- Tablets at each table, which customers can use to place food orders.
- A drink-dispensing system where customers can self-serve various beverages.
- A tablet in the kitchen where food orders are received and managed.

The core functions for the website client include:

- Displaying the current food menu.
    - This menu is manually updated.
- Displaying the current drink menu.
    - This menu should be automatically updated based on the current status provided by the drink-dispensing system.
- Allowing customers to place take-away food orders.

The core functions for the table tablet clients include:

- Displaying food and drink menus (same functions as those used on the website).
- Allowing customers to place eat-in food orders.
- Displaying the status of food orders, as well as displaying notifications about orders.
- Allowing customers to call a server.

The core functions for the drink-dispensing system client include:

- Displaying the current drink menu.

- Allowing customers to self-serve drinks.
    - Customers should be able to either pay for an individual drink or add the cost to their eat-in menu.
    - When a drink order is placed, the hardware should dispense a specific quantity of the beverage to the customer's glass.
    - If the beverage cannot be dispensed, the customer should not be charged.
    - Once dispensed, the menu status should be updated (remaining quantity updated, or drink is removed if the quantity is now 0).

The core functions for the kitchen tablet client include:

- Displaying all current orders.
- Updating the status of an order.
- Sending custom notifications about orders to a table.
- Updating the food menu.

You may add additional functionality or make decisions on how the functionality described above is implemented, as long as these decisions fit the overall purpose of the system and are well-explained. In any case, state any assumptions that you make and describe any functionality decisions you create.

In the following problems, you will develop requirements and scenarios for each of the four quality attributes described in class.

- Use the quality scenario format from Lecture 3 and Exercise Session 1: Overview, System State, Environment State, External Stimulus, Required System Response, Response Measure.
    - The handout for Exercise Session 1 offers an overview of these fields, and both the lecture and exercise session offer examples of scenarios.
    - Scenarios should have a single clear stimulus and specific and measurable response measures related to the quality of interest.
- The requirements should describe the expected quality of a function-under-consideration under a set of clear conditions. They should be specific and testable.
    - **Bad:** "The system shall have a ROCOF under 2 failures in a five minute period."
    - **Still bad:** "The payment function shall have a ROCOF under 2 failures in a five minute period."
    - **Good:** "Under both normal (500 payment requests submitted within a five minute period) and heavy (defined as 10000 payment requests submitted within a five minute period) load, fewer than two requests submitted to the payment function during this period shall fail."

Please come up with your own original requirements and scenarios. Do not take the examples from class and lightly adapt them.

Problem 1 - Reliability (25 Points)

- Specify two **reliability** requirements for selected functionalities of this system.
- For each requirement, develop a **reliability** scenario that demonstrates that this requirement is met.

Problem 2 - Availability (25 Points)

- Specify two **availability** requirements for selected functionalities of this system.
- For each requirement, develop an **availability** scenario that demonstrates that this requirement is met.

Problem 3 - Performance (25 Points)

- Specify two **performance** requirements for selected functionalities of this system.
- For each requirement, develop a **performance** scenario that demonstrates that this requirement is met.

Problem 4 - Scalability (25 Points)

- Specify two **scalability** requirements for selected functionalities of this system.
- For each requirement, develop a **scalability** scenario that demonstrates that this requirement is met.