SWE Bonus Sheet (UML)

Question No. 1

Company X will develop a Festival ticket system (FTS) to be used to manage tickets and access to festivals in the town of Trondheim. The system includes a server computer and software to manage the operations of the system. The system provides a mobile interface to enable festival visitors to buy tickets. Moreover the system provides a mobile interface to enable festival personnel to check tickets at the entrance. Last, the system provides a web based interface for administration functions, such as declaration of new festivals and statistics generation. The most basic functions are to handle ticket sales and to check tickets at entrance. When a visitor wants to buy a ticket, he logs in the system, the FTS starts a transaction, checks that there are enough available tickets for the given festivals. When the sale transaction is over, the visitor can pay in cash or credit card. After the payment is successful, the visitor gets a Quick Response (QR) code sent to his mobile. Only credit card payment is supported. Only mobile based sales are supported. When a visitor arrives at a festival, the festival worker in charge uses his mobile to scan the QR CODE of the visitor, and then FTS will retrieve the name of the visitor from the backend visitor system and interact with ticket system to update the number of visitors to this festival. The users of the FTS system are festival visitors, festival workers, and the administrator. The administrator can access the system management functions of the FTS system including festival management and security configuration.

I. Make use case diagrams for all functions in the system

II. Given that one of the main functional requirements is “F1 Buy Ticket” and one of its sub functional requirements is “F1.1 Retrieve availability and price of festival”, list the other functional requirements of the system.

III. Draw a sequence diagram for the scenario of buying one ticket.

Question No. 2

Consider the FTS system in Question No. 1:

I. Briefly describe the main phases you plan to test the system.

II. Write high level Test cases (Test data, preconditions and expected results) for testing the functional requirement Retrieve availability and price of festival

Question No. 3

Consider the software to be embedded in a nebulizer pump used by children who receive asthma treatment at hospitals or day care centers. The software system is connected to the internet and it can access parameters such as day temperature.

Before use, the software system must be set up by personnel with parameters about the child, such as weight, age, seriousness of sickness. The system calculates the amount of medicine required to be delivered and the duration of the treatment based on information about the child and the temperature of the day. It sends signals to a micro-pump to deliver the correct dose of medicine. During the treatment an adult must be with the child

I. Identify the main three stakeholders
II. Define the main Use Case Diagram for the system.
III. Define the main classes and the main class diagram.

Question No. 4

Your customer is tired of his employees who keep coming late from coffee brakes because it takes them ages to prepare and drink their coffee and to clean their equipment afterwards.

He asks you to design (***Draw use case diagram***) the prototype of a new revolutionary coffee machine that is both appealing to the eye and yet easy enough to handle even for an IT specialist.

During a brainstorming session on functional requirements, among the first use cases that are brought up there are things like prepare, serve, and cleanUp. While appearing simple at first glance, it soon turns out that the design must pay attention to a lot of details, so for each of the above mentioned use cases there will probably be several included use cases like insertCoffee, or addMilk. Also bear in mind that the coffee machine needs to be refilled, maintained. There might be power brakes or problems with the water supply. Try to cover as many boundary conditions as possible to ensure the office keeps running.

Do not spend too much time on the individual use cases. The emphasis of this task lies on the syntax and semantics of the UML use case diagram. Your diagram should contain examples of <<includes>> and <<extends>> relations. An example of generalization would be nice.