UNIVERSITY OF Southampton

Michael Butler
mjb@ecs.soton.ac.uk
Dana Dghaym
D.Dghaym@ecs.soton.ac.uk
Thai Son Hoang
T.S.Hoang@ecs.soton.ac.uk
Yvonne Howard
ymh@ecs.soton.ac.uk

COMP1216. Software Modelling and Design (2019-20)

Solution Sheet 1: Requirements Analysis

Issue date: 3 February 2020

Part 1. An Email System

The following is a simple requirements specification for an *e-mail system*.

1. Users can send emails to each other
2. A user should be able to read emails that have been sent to them
3. The average delivery time for emails should be less than 5 seconds
4. Emails are sent and received via mail servers
5. Mail servers are connected via SMTP messages
6. Users must provide a password to access their email
7. Inform the user if a message transmission fails
8. In case of transmission failure, sending should be re-tried after 4 hours
9. Messages can be composed in plain text or in formatted text
10. The format of an SMTP message is
    - 1 or more headers (recipient, return address, routing, etc)
    - Body of message in text or html

Question 1. Requirements Classification

Categorise these into *Functional* (**F**), *Non-Functional* (**NF**), and *Design* (**D**) requirements.

Solution:

1. Users can send emails to each other (**F**)
2. A user should be able to read emails that have been sent to them (**F**)
3. The average delivery time for emails should be less than 5 seconds (**NF**)
4. Emails are sent and received via mail servers (**D**)
5. Mail servers are connected via SMTP messages (**D**)
6. Users must provide a password to access their email (**F**)
7. Inform the user if a message transmission fails (**F**)
8. In case of transmission failure, sending should be re-tried after 4 hours (**F**)
   Note: Ideally this should be seperated into different requirements (1) for resending (**F**) and (2) after 4 hours (**NF**).
9. Messages can be composed in plain text or in formatted text (**F**)
10. The format of an SMTP message is (**D**)
    - 1 or more headers (recipient, return address, routing, etc)
    - Body of message in text or html

Part 2. A Train Ticket System

A system must be specified for the automated purchase of train tickets from a ticket distributor. It is possible for the traveller to buy single or return tickets to available destinations, as well as weekly and monthly season tickets. The traveller will interact with the machine to specify ticket type, select destination, select payment mode (cash or credit card). A ticket purchase transaction may fail for various reasons: the distributor is out of change, out of ticket paper, credit card fails to validate, etc.

Question 2. Defining Scope

Identify the scope of the ticket system, i.e., *Need*, *Goals*, *Business Case*, *Stakeholders*, *High-level operational concepts*, etc.

Solution:
Below is an example solution

- *Need*: our current manual process for purchasing ticket is time consuming and not effective.

- *Goal*: provide an automated system that allow the customers to purchase train tickets quickly and easily.
- *Business case*: reduce operational cost by requiring less staff
- *Stakeholders*: travellers, maintenance staff, IT department, banks
- *High-level operational concepts*:
  - The staff can set ticket prices.
  - The staff can add/remove destinations.
  - Travellers can interact with the machine to specify ticket type, destination, and payment method.
  - Travellers can add more tickets to their basket
  - Travellers can checkout and pay using different methods
  - Travellers can cancel their order anytime before payment is confirmed.

Question 3. Operational Scenarios

Specify two operational scenarios for this system for

1. a successful ticket purchase, and
2. a failed ticket purchase.

Solution:

1. **Scenario 1. The customer successfully buy a weekly ticket**
   - The customer selects "Buy weekly card" and is taken to the "Select destinations" screen
   - The customer selects a destination and is taken to the "Basket" screen
   - The customer selects "Pay" and is take to the "Select payment type" screen.
   - The customer selects "Credit Card" and enters the card
   - The distributor validates the card and requests the PIN
   - The customer enters a PIN and the distributor validates it by reference to the bank
   - The distributor takes payment and returns the credit card
   - The weekly season card is printed and the customer takes it
2. **Scenario 2. The customer failed to by a one-way ticket**
   - The customer selects "One way ticket" and is taken to the "Select destinations" screen
   - The customer selects the destination and is taken to the "Basket" screen
   - The customer selects "Pay" and is take to the "Select payment type" screen.
   - The customer selects "Cash"

- The distributor reports that insufficient change is available, and offers "Continue" and "Cancel" options
- The customer cancels (and goes to the ticket office to buy the ticket).

Note:

- Bullet point or list style is better for scenarios
- Make sure to document each interaction step on its own. This clarifies the flow of cause and effect in the user-system interaction. For example "The customer chooses a return ticket to some destination" is too abstract, it loses the two interactions steps "Choose destination" and "Choose ticket type".

Question 4. Writing Requirements

Based on the scenarios in Question 3, write down the list of requirements for the ticket system. Label and number the requirement accordingly, e.g., **ASM** for assumptions, **FUN** for functional requirements.

Solution:
An example set of requirements is as follows. We also use **DES** for design requirements.

- **FUN1**: The system shall provide a "Select ticket-type" screen with the options for *one-way*, *return*, *weekly*, *monthly* tickets.
- **FUN2**: Once the ticket-type is selected, the system shall provide a screen for selecting destination.
- **DES1**: The destination can be chosen by typing from a keyboard or select from a list.
- **FUN3**: Once the destination is chosen, the ticket is added to the basket, and the system shall provide a screen show the current basket.
- **FUN4**: From the "Basket" screen, the traveller can choose to add more tickets or pay for the current basket.
- **FUN5**: If the traveller choose to add more tickets, the "Select ticket-type" screen is showed.
- **FUN6**: If the traveller choose to pay, the "Select payment type" screen is showed with option for *cash*, *credit card*.
- **ASM1**: Travellers may have credit cards, each with a PIN, which can be verified by banks.
- **ASM2**: Travellers will keep the PIN of their credit card secret.
- **FUN7**: The system can validate credit cards inserted by the traveller.
- **FUN8**: Once the credit card is validated, the system will ask the traveller for the PIN
- **FUN9**: Once the credit card and the PIN is verified by the bank, the system will take the payment and return the card.

- **FUN10**: Once payment is accepted, the system will issue all the tickets in the basket.
- **FUN11**: All screens have "Cancel" button to clear the "Basket" and return to the "Select ticket-type" screen.

Note that the above list of requirements are incomplete, e.g., we have not yet had any requirements for the staff to update the destination or the ticket prices.