Problem Sheet 3. Requirements Engineering

Problem 1. You're creating a mental healthcare patient information system. Who might the stakeholders be?

Problem 2. Analyze each of the following (separate) software requirement statements for clarity, consistency, verifiability, unambiguity, and completeness. Provide a reformulation of the following software requirement statements to improve them.

1. The Dropbox software shall do its best to ensure fast cloud uploads.
2. When the primary unit is lost, the secondary unit takes over.
3. The system shall purge state control records and files that are older than the retention period.
4. When performing calculations the software shall produce correct results.
5. The output of the program shall usually be given within 10 seconds.

Problem 3. Which of these are functional software requirements?

1. Users of the library will be either normal or staff
2. A user will be able to borrow a book
3. A staff person will be able to borrow a book
4. The library contains one million books
5. If a user asks a book that has been borrowed, her request shall be inserted in a waiting list
6. Staff shall have no priority in borrowing books

Problem 4. Which of these are functional requirements?

a) A person can enroll in a course
b) Only 10 persons max can enroll to this course
c) A student is a person
d) A course is taught by a professor
e) A professor is not a student
f) Each course terminates with an exam
g) Every exam produces a ranking of all participants

Problem 5. Discover ambiguities or omissions in the following statement of requirements for part of a ticket-issuing system.

An automated ticket-issuing system sells rail tickets. Users select their destination and input a credit card and a personal identification number. The rail ticket is issued and their credit card account charged. When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select a destination. Once a destination has been selected, users are requested to input their credit card. Its validity is checked and the user is then requested to input a personal identifier. When the credit transaction has been validated, the ticket is issued.

Problem 6. Rewrite the above description using the structured approach described in this chapter. Resolve the identified ambiguities in an appropriate way.

Problem 7. Write a set of non-functional requirements for the ticket-issuing system, setting out its expected reliability and response time.

Problem 8. Write plausible user requirements for the following functions, where natural language descriptions are presented in a standard format:

- An unattended petrol (gas) pump system that includes a credit card reader. The customer swipes the card through the reader then specifies the amount of fuel required. The fuel is delivered and the customer's account debited.
- The cash-dispensing function in a bank ATM.
- The spelling-check and correcting function in a word processor.

Problem 9. Using your knowledge of how an ATM is used, develop a set of use cases that could serve as a basis for understanding the requirements for an ATM system.

Problem 10. Suppose you want to build a program called TimeShifter to upload and download files at scheduled times while you're on vacation. The following list shows some of the application's requirements. For this exercise, list the audience-oriented categories (business, user, functional, non-functional, or implementation) for each requirement. Are there requirements in each category?

1. Allow users to monitor uploads/downloads while away from the office.
2. Let the user specify website log-in parameters such as an Internet address, a port, a username, and a password.
3. Let the user specify upload/download parameters such as number of retries if there's a problem.

4. Let the user select an Internet location, a local file, and a time to perform the upload/download.
5. Let the user schedule uploads/downloads at any time.
6. Allow uploads/downloads to run at any time.
7. Make uploads/downloads transfer at least 8 Mbps.
8. Run uploads/downloads sequentially. Two cannot run at the same time.
9. If an upload/download is scheduled for a time when another is in progress, it waits until the other one finishes.
10. Perform scheduled uploads/downloads.
11. Keep a log of all attempted uploads/downloads and whether they succeeded.
12. Let the user empty the log.
13. Display reports of upload/download attempts.
14. Let the user view the log reports on a remote device such as a phone.
15. Send an e-mail to an administrator if an upload/download fails more than its maximum retry number of times.
16. Send a text message to an administrator if an upload/download fails more than its maximum retry number of times.

Problem 11. Using the Moscow model, brainstorm what features are a must, should, could, or won't for software such as Dropbox, where the goal of the system is to synchronize files between devices.