

## Tutorial 1 - Requirements Capture *Sample Solution*

### Possible Stakeholders

(“Stakeholder is anyone who should have some direct or indirect influence on the system requirements.” – Sommerville)

Management of the ferry company

Management of the travel agencies

Travel Agents (the end-users)

Staff of the (ferry company) Head Office

Staff of the ferry company (handling tickets, possibly)

Customers of the travel agencies are unlikely to be stakeholders unless we choose to consult them about the requirements for the new system. We could survey the customers to see what they would like and we might find additional features such as e-mail confirmation of bookings.

They are most likely to ask for a Web-based booking system but this is outside the scope of this exercise!

### Functional Requirements

Answer timetable queries

Answer enquiries about possible bookings

Make provisional bookings and provide printed quotation

Make firm booking (receive deposit) and issue booking confirmation slip

Print tickets (after receiving full payment)

Update timetable {Do we still need this? Possibly, if we keep local copy?}

How payments are handled is not clear from the description and would require further investigation.

### Non-functional requirements

Hardware **compatibility** - the travel agencies (management) want to use existing hardware platforms.

External system **compatibility** – the new system must conform to the interfaces of the central booking system for exchanging information.

**Usability** - the need for a simple to use and robust interface with guidance and on-line help (will be associated with all functions that have interactive interfaces used by Travel Agents).

**Reliability** of communications between the central booking system and the travel agencies.

**Security** of data transmission from central booking system to travel agencies.

The definition of a *larger* travel agency could be regarded as a non-functional requirement as it will have implications for the continued support of the existing head-office system in parallel with the new system.

### Missing requirements

For **functional** requirements, probably the most obvious deficiency is the lack of any management reporting or auditing functions. The management of the Travel Agency will want to know how much business is going to the Ferry Company and conversely the Ferry Company Management will want to know about the business coming from the different travel agencies. We should also pick up the need

for standard functions such as cancel booking or amend booking (at various stages in the booking process) as we develop more detailed requirements.

**Non-functional** requirements. No mention of security at the travel agents terminal, presumably password protection or swipe card would be required (or any other affordable security mechanism!). Training needs have not been mentioned or any plan for phasing in the new system. There is also the question of performance of the system, for example the number of concurrent users that should be supported (peak load) and the maximum response time to an individual request (a good example of a trade-off here between the number of users that can be supported and individual response time).

## Testing the Requirements

*I have put the emphasis on non-functional requirements; there should not be much difficulty with the testing of the functional requirements.*

Some of the non-functional requirements are simply *constraints* on what can be done. The hardware compatibility will constrain the whole development process, while the external interface compatibility will constrain parts of the design.

Security of communications is an interesting problem. The best approach here is probably to choose an industry standard for encryption and get an implementation off the shelf.

Other non-functional requirements need to be developed into more detailed measurable requirements. For example, the **reliability** might be expressed as: the percentage of working hours that the connection is available (up time); mean time between failures of the connection; maximum time permitted for re-connection after failure (alternatively, mean time to repair connection failure). Connection failure covers both loss of the physical connection and failure of the Head Office system. To provide some service during down time of connection, we might keep a local copy of the timetable on-line that is automatically updated from the Head Office.

There are lots of possible discussion points about **usability**. How experienced are the users? How often are they likely to use the ferry company interface? What types of interfaces are they familiar with? Which current interface(s) do they prefer? Should we model our interface on one of the existing interfaces they use? How are we going to measure the usability of the interface we do deliver (time to learn how to use it, error rates after learning period, evaluation using think-alouds or other forms of feedback, ...)?

System performance could be tested by using artificial loads, with pseudo terminal input, (stress testing with maximal loads) and, later, monitoring the actual performance of the system in normal operation.

*(Note. This is not **the** solution to this exercise. Given the vagueness of the original problem definition, there are always other ideas that could reasonably be added to this.)*