

# CHAPTER 1: CLIENT REQUIREMENTS

## Objectives

The objectives are:

- Introduction
- Diagnosis – Executive Summary
- Analysis – Functional Requirements
- Other Requirements
- Data Model
- Project Plan

## Introduction

Ideally, the first step in solution development is an analysis of specifications, functional requirements, data models, and use cases. Proceeding with implementation without these materials is likely to result in an unpleasant experience, both for developer and customer. Planning documentation is the basis of the contract between developer and customer, and use cases help reveal flaws and/or deficiencies that may be present in the design, not just at the outset, but as the solution is implemented.

The diagnosis section of this chapter provides the "executive summary" of the business case, including the client's profile and a high-level description of their needs. The Analysis sections describe the client's specific requirements for the system, including the data models based on those requirements and the basic project implementation plan.

In subsequent chapters, the analysis, design, and development phases are broken down in the chapter exercises. These exercises constitute separate pieces of the development project, but they are inter-related and therefore dependencies do exist.

After the development is completed, Chapter 10 discusses deployment issues, and Chapter 11 addresses SQL Server performance optimization.

### Diagnosis – Executive Summary

This course is built around the idea that the reader is a Certified Microsoft Dynamics™ NAV Developer working for a Microsoft Certified Dynamics Partner. The project that has been assigned is for CRONUS International Training Academy, a software training center that is a branch of CRONUS International Ltd. Due to significant growth, they need a new computer-based system that allows them to both store and integrate all of their seminar, instructor, customer, and financial information in one solution.

The client currently uses a full suite of Microsoft Dynamics NAV granules under the parent company CRONUS International Ltd. To take advantage of their investment and of the existing functionality and flexibility of Microsoft Dynamics NAV, they have decided to add a customized seminar management module to their current solution.

This new module should allow them to:

- Track their master data
- Register participants in their seminars
- Create invoices for customers
- Have an overview of their statistics

The preliminary analysis and design of the processes and requirements has already been done and is included in this training manual.

If the client is satisfied with the seminar management module, this type of system could be sold to other training academies. The solution must therefore be developed in such a way that other companies could also use it.

### Analysis – Functional Requirements

The client has defined their requirements for the new seminar management module by providing the following description of how they run their training academy.

#### Seminars

The CRONUS training department holds several different seminars. Each seminar has a fixed duration, and a minimum and maximum number of participants. In some cases seminars can be overbooked, depending on the capacity of the assigned room. Each seminar can be cancelled if there are not enough participants. The price of each seminar is fixed.

To take advantage of the current Job functionality in Microsoft Dynamics NAV, each seminar will be defined as a job. When a seminar is completed, the seminar should be posted as a job, with additional seminar-specific information.

Each seminar is held in a seminar room. Some are held in-house and some are held off-site. If a seminar takes place in-house, a room must be assigned. For off-site rooms, the rental rate must be tracked as well.

### **Instructors**

Each seminar is taught by an instructor, who is a CRONUS employee. To make use of our existing resource information, each instructor must be set up as a Resource in Microsoft Dynamics NAV.

### **Participants**

Seminar participants come from a company that is set up as a customer in Microsoft Dynamics NAV, but must be handled separately from the customers. Every customer can register several participants for a seminar but participants cannot be registered unless they are connected with a customer. This is necessary to invoice customers for the participation at seminars.

The system should gather additional participant information, such as the number and names of previously attended seminars.

### **Registration**

Customer can register one or more participants for a seminar. Each registration is assigned a job number. It must be possible to assign additional expenses to an instance of a seminar, such as catering expenses or equipment rentals. The registration information must include how the seminar should be invoiced (for example, whether to include expenses or catering).

Additional comments for each seminar can be entered to allow for things like necessary equipment or other particular requirements.

### **Invoicing**

When each seminar is finished, the customers with participants should be invoiced. This should be done by project and resources.

### **Reporting and Statistics**

The customer requires that a number of reports and statistics be available. These include:

- A list of the participants registered for a seminar.
- A list of the participants that have successfully completed each seminar, for the purposes of providing certificates. (The certificates themselves will also be produced by the system.)

- The total costs for each seminar, broken down according to what is and what is not chargeable to the customer.
- Statistics for different time periods, for example, for a month, for last year, for this year, and up to the current date.

### Interfaces

The solution should allow e-mail notification to be sent to the customer's participants in several situations, such as registration confirmation. Also, the participant list for a seminar should be exportable as an XML file.

### Dimensions

Analysis features should be provided by extending the standard customer dimension functionality in the seminar module. Dimensions should be available for the master files, registrations, posting and invoicing.

### Calendars

A calendar system that provides an overview of the seminar dates to help in seminar planning. It should allow seminars to be viewed by date and to set filters to see the overview for seminars with a specific seminar status, seminar room or instructor.

### Other Requirements

To make the solution as user-friendly for the client as possible, keep the following requirements in mind throughout development:

**Easy to learn:** The seminar management module must be easy to understand, and the terminology and symbols must be consistent with the rest of the program. This means that if the user knows how to use other Microsoft applications, they will also be able to intuitively learn our solution.

**Efficient:** Experienced users should be able to work with the program efficiently. This means, for instance, that the most frequently used functions should be accessible from the keyboard. It should be possible to efficiently use both the mouse and the keyboard.

**Clarity:** The user interface must be intuitive so that the least-experienced user can easily understand how the program functions.

**Good error reporting:** The program must be built so that there are as few opportunities for error as possible. Error messages must explain the cause of the error and provide a suggestion as to how the user can correct the error.

**Design consistency:** Because the solution for the seminar management module could be sold as an add-on to other customers, it should adhere to the guidelines contained in the Application Designer's Guide.

## Data Model

Figure 1-1 shows the data model that encapsulates client's specification of the functional requirements.

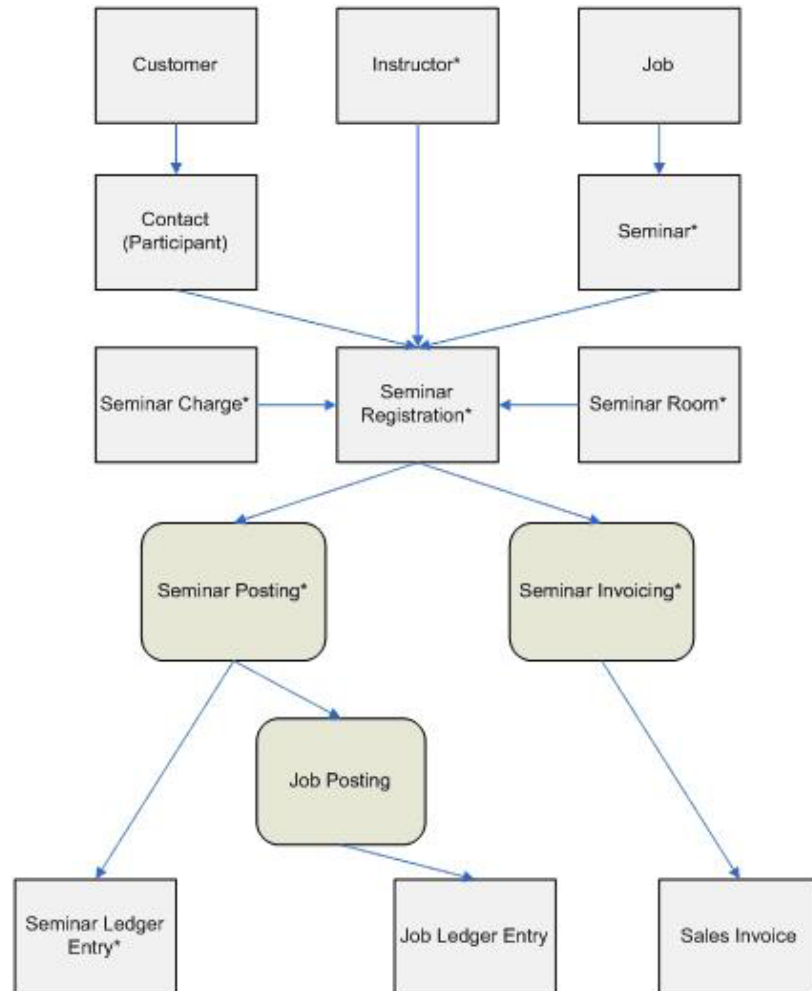


FIGURE 1-1: THE DATA MODEL

### Project Plan

As in any other project, our business case must be broken down into tasks, which make up the exercises in each chapter. There are dependencies among the exercises and the technical complexity of the project increases as development progresses.

The structure of the exercises reflects the principles of the implementation methodology. Diagnosis and analysis phases therefore precede the development and testing of the individual tasks. The end result of a chapter's exercises will be a deliverable for that particular stage of the project.

The tasks for this project are:

- Master Tables and Forms (Chapter 2)
- Registrations (Chapter 3)
- Posting (Chapter 4)
- Integration (Chapter 5)
- Reporting (Chapter 6)
- Statistics (Chapter 7)
- Dimensions (Chapter 8)
- Interfaces (Chapter 9)

### Conclusion

Throughout this course, each implementation phase, or exercise, is preceded by an analysis and design phase. This practice is meant to minimize the time it takes to implement a solution and optimize the solutions effectiveness.

### Quick Interaction: Lessons Learned

Take a moment to write down three Key Points you have learned from this chapter:

1.

---

---

---

---

2.

---

---

---

---

3.

---

---

---

---

